

Adaptive, linear, subspatial projections  
for invariant recognition of objects  
in real infrared images

Michael Howard William Smart



Thesis submitted for the degree of  
Doctor of Philosophy  
The University of Edinburgh  
February 1998



## **Abstract**

In recent years computer technology has advanced to a state whereby large quantities of data can be processed. This advancement has fuelled a dramatic increase in research into areas of image processing which were previously impractical, such as automated vision systems for, both military, and domestic purposes.

Automatic Target Recognition (ATR) systems are one such example of these automated processes. ATR is the automatic detection, isolation and identification of objects, often derived from raw video, in a real-world, potentially hostile environment. The ability to rapidly, and accurately, process each frame of the incoming video stream is paramount to the success of the system, in order to output suitable actions against constantly changing situations.

One of the main functions of an ATR system is to identify correctly all the objects detected in each frame of data. The standard approach to implementing this component is to divide the identification process into two separate modules; feature extraction and classification. However, it is often difficult to optimise such a dual system with respect to reducing the probability of mis-identification. This can lead to reduced performance. One potential solution is a neural network that accepts image data at the input, and outputs estimated classification. Unfortunately, neural network models of this type are prone to misuse due to their apparent black box solutions.

In this thesis a new technique, based on existing adaptive wavelet algorithms, is implemented that offers ease-of-use, adaptability to new environments, and good generalisation in a single image-in-classification-out model that avoids many of the problems of the neural network approach. This new model is compared with the standard two stage approach using real-world, infrared, ATR data.

Various extensions to the model are proposed to incorporate invariance to particular object deformations, such as size and rotation, which are necessary for reliable ATR performance. Further work increases the flexibility of the model to further improve generalisation. Other aspects, such as data analysis and object generation accuracy, which are often neglected, are also considered.

# **Declaration**

I declare that this thesis has been completed by myself and that, except where indicated to the contrary, the research documented is entirely my own.

Michael H. W. Smart

## Acknowledgements

Well it's 3:38p.m. on a rather overcast February Sunday in Edinburgh, two million, three hundred thousand, six hundred and eighteen minutes past the start of this fifty thousand, four hundred and fifty four word thesis. By my reckoning, which I hasten to add should be treated with extreme caution at this moment in time, gives me an average typing speed of approximately 45 minutes per word and, more than likely, rule me out of secretarial school. Anyway, I stray. Who would I like to thank? Well quite a few people in fact: my parents and family, of course, for their unceasing support and devotion; to all the members, past and present, of the Integrated Systems Group for their kindness and help; to my friends for, well, simply being there; and to the members of staff both here at the University of Edinburgh Engineering Department and at British Aerospace Systems and Equipment for making this thesis possible.

Thanks to Uncle Robin for, not only providing excellent thesis reviews, but for simply being Uncle Robin. To Emma a big hug for pushing me on at the very last. Richard, thanks for all your idea's and thought's during the project, for making Plymouth tolerable and for some, let's say, more than interesting conversations. To Andy for the thesis reviews, for sharing an office, and, of course, the citrus experience. To all the neural-teenies, thanks.

For some very happy memories during the past few years here in Edinburgh thanks to Mark and Alistair, Camilla, Doug, Mandy, Drew and Mu, Emma, Masa, Karen, Adrian, Richard, Deborah, Paul and Mick, and lastly but far from least, Nadia.

I would like to thank Professor Alan Murray for his supervision and for the opportunity to study here in Edinburgh and to Dave Edmondson and Mike Green at BASE for their assistance in the project, as well as the cash of course. To EPSRC for funding the project and myself for three years, and to Belhaven breweries for de-funding me over the past three years. Finally, Gertrude, Basil, and Daphne it was nice but lets hope we don't meet again. Well that's about it I think, so a final thank you to one and all. Corking.



---

# Contents

---

<b>List of Figures</b>	<b>x</b>
<b>List of Tables</b>	<b>xvi</b>
<b>1. Thesis overview</b>	<b>1</b>
1.1 Thesis background . . . . .	1
1.2 Project aims . . . . .	4
1.3 Thesis outline . . . . .	5
1.4 Areas of contribution . . . . .	7
1.5 Review . . . . .	8
<b>2. Automatic target recognition and the OSTRICH project</b>	<b>9</b>
2.1 Recognition . . . . .	9
2.2 Automatic target recognition (ATR) . . . . .	10
2.3 Object classification . . . . .	15
2.3.1 Linear classifiers . . . . .	17
2.3.2 K-nearest neighbour classifiers . . . . .	18
2.3.3 Multi-layer perceptron classifiers . . . . .	19
2.3.4 Other classifiers . . . . .	22

Contents	iv
2.4 The OSTRICH ATR system . . . . .	24
2.4.1 System overview . . . . .	24
2.4.2 Original classification module . . . . .	26
2.4.3 Problems with the original module . . . . .	26
2.4.4 A solution . . . . .	27
2.5 Databases available . . . . .	28
2.5.1 Simple test data . . . . .	28
2.5.2 NIST digit database . . . . .	28
2.5.3 Forward-looking infrared (FLIR) images . . . . .	29
2.6 A nontrivial problem . . . . .	31
2.7 Review . . . . .	34
<b>3. IR object segmentation, analysis and preprocessing</b>	<b>35</b>
3.1 Object segmentation . . . . .	37
3.1.1 Sobel-based segmentation . . . . .	38
3.1.2 Hand labelling of segmented objects . . . . .	41
3.1.3 Confirmation of the labelled data . . . . .	44
3.1.4 Seascape segmentation problems . . . . .	45
3.1.5 Other segmentation techniques . . . . .	48
3.2 Object analysis . . . . .	49
3.2.1 Bounding box analysis . . . . .	49
3.2.2 Outline analysis . . . . .	51
3.2.3 Binary mask analysis . . . . .	53

3.2.4	Grey-level analysis . . . . .	54
3.2.5	Abstract level analysis . . . . .	58
3.2.6	Analysis conclusions . . . . .	59
3.3	Object preprocessing . . . . .	61
3.4	Review . . . . .	63
4.	<b>Feature extraction and classification</b>	<b>65</b>
4.1	Feature extraction . . . . .	67
4.1.1	Determinedness . . . . .	67
4.1.2	Statistical features . . . . .	68
4.1.3	Linear spatially-mapped features . . . . .	68
4.2	Feature analysis . . . . .	79
4.2.1	Feature separability . . . . .	79
4.2.2	3D object rotation . . . . .	81
4.2.3	Outliers . . . . .	82
4.2.4	Multi-modality . . . . .	83
4.2.5	Feature confidence . . . . .	85
4.2.6	Normalisation . . . . .	85
4.3	Preliminary classification . . . . .	86
4.3.1	Classifier experimental setup . . . . .	86
4.3.2	Classifier results . . . . .	88
4.3.3	Comments . . . . .	93
4.4	Feature Selection . . . . .	94

4.4.1	Using <i>a priori</i> knowledge . . . . .	94
4.4.2	Individual feature selection . . . . .	95
4.4.3	Subset selection . . . . .	100
4.4.4	Reconstruction . . . . .	103
4.4.5	Other feature selection techniques . . . . .	104
4.5	Analysis . . . . .	104
4.6	Review . . . . .	107
<b>5. Adaptive kernel neural networks</b>		<b>109</b>
5.1	Kernel feature extraction . . . . .	110
5.2	Adaptive kernel feature extraction . . . . .	112
5.2.1	Adaptive wavelets . . . . .	112
5.2.2	Error derivatives . . . . .	113
5.2.3	Constraints on the form of $\psi$ . . . . .	116
5.2.4	Kernel selection . . . . .	116
5.3	Adaptive kernel experiments . . . . .	118
5.3.1	Linear classification . . . . .	118
5.3.2	Nonlinear classification . . . . .	139
5.4	Review . . . . .	140
<b>6. Invariance with adaptive kernel networks</b>		<b>142</b>
6.1	Invariance . . . . .	142
6.2	Polar image representation . . . . .	145

- 6.3 Review of current invariant techniques . . . . . 146
- 6.4 Invariance through training . . . . . 147
- 6.5 Invariance through structure . . . . . 148
- 6.6 Invariance through feature extraction . . . . . 149
  - 6.6.1 Translation . . . . . 150
  - 6.6.2 Scale . . . . . 150
  - 6.6.3 Rotation . . . . . 151
- 6.7 Digital approximation . . . . . 159
- 6.8 Classification of fixed RI features . . . . . 159
- 6.9 Adaptive invariant techniques . . . . . 162
  - 6.9.1 Adaptive complex kernel feature extraction . . . . . 163
  - 6.9.2 RI through  $\theta$  normalisation . . . . . 167
  - 6.9.3 Linear adaptive kernel . . . . . 169
  - 6.9.4 Nonlinear adaptive kernel . . . . . 170
- 6.10 Review . . . . . 173
- 7. Integration into the ATR environment 174**
  - 7.1 The effects of rogue data . . . . . 176
    - 7.1.1 Clutter . . . . . 176
    - 7.1.2 Occlusion . . . . . 177
    - 7.1.3 Segmentation failure . . . . . 178
  - 7.2 The identification of rogue data . . . . . 182
    - 7.2.1 Classifier outputs and *a posteriori* probabilities . . . . . 182

Contents	viii
7.2.2 $C + 1$ classification . . . . .	183
7.2.3 Novelty classification . . . . .	188
7.2.4 Identification conclusions . . . . .	194
7.3 The separation of rogue data . . . . .	195
7.4 Database adaptability . . . . .	196
7.4.1 Database description . . . . .	196
7.4.2 Database results . . . . .	197
7.4.3 Database conclusions . . . . .	198
7.5 Review . . . . .	200
<b>8. Conclusions</b>	<b>201</b>
8.1 Summary of work completed . . . . .	201
8.2 Analysis of completed aims . . . . .	202
8.3 Scope for future work . . . . .	206
8.4 Final comment . . . . .	207
<b>Bibliography</b>	<b>208</b>
<b>A. Optimisation techniques</b>	<b>217</b>
A.1 Simple descent methods . . . . .	217
A.2 The Simplex Method . . . . .	218
A.3 Conjugate gradient optimisation . . . . .	220
<b>B. Publications</b>	<b>222</b>

Biological and Artificial Computation . . . . . 223

Proceedings of the SPIE . . . . . 233

---

# List of Figures

---

2-1	A common serial ATR system. . . . .	11
2-2	A section of the electromagnetic spectrum. . . . .	12
2-3	A comparison of modern ATR sensors. . . . .	14
2-4	Optimal quadratic decision boundary for a two class problem. . . . .	18
2-5	Architectural diagram of an MLP. . . . .	19
2-6	The logistic function commonly used as the MLP non-linearity, $\varphi$ . . . . .	20
2-7	Too little, and too much, flexibility resulting in model bias and variance. . . . .	21
2-8	System overview at $n^{th}$ time step . . . . .	25
2-9	NIST: Examples from the fl3 database. . . . .	29
2-10	Seascape: Typical infrared database scene. . . . .	30
2-11	Seascape: Example of the required classifier response. . . . .	31
2-12	Seascape: Different types of information . . . . .	32
3-1	OSTRICH: Creation of an object database. . . . .	36
3-2	OSTRICH: Sobel segmentation module. . . . .	39
3-3	Sobel masks for generating $H_x$ and $H_y$ . . . . .	40
3-4	Horizon filter. . . . .	40
3-5	Seascape: Classification tree for the object databases. . . . .	42



<i>List of Figures</i>	xi
3-6 Seascape: Subclass populations of all 4003 segmented objects. . . . .	44
3-7 Seascape: An example of the non-closure segmentation problem. . . . .	45
3-8 Seascape: Review of the 4003 segmented objects. . . . .	47
3-9 Seascape: Distribution of well-segmented, object sizes. . . . .	50
3-10 Seascape: Box-plots for bending energy and compactness in each class. . . .	52
3-11 Seascape: Curvature and $(\rho, \theta)$ plot for a sailboat outline. . . . .	53
3-12 Seascape: Foreground grey-level histogram for a sailboat, motor boat and buoy.	55
3-13 Seascape: Class normalised centroid distributions. . . . .	56
3-14 Seascape: Angles of minimum asymmetry (maximum symmetry.) . . . . .	57
3-15 Seascape: Rose diagram showing directional populations of sailboat class. . .	59
3-16 Re-sampling examples . . . . .	61
3-17 Seascape: Grey-level histograms for a typical sailboat. . . . .	62
3-18 OSTRICH: The preprocessing system. . . . .	63
4-1 OSTRICH: Feature extraction and classification stages. . . . .	66
4-2 NIST: Effects of zoning upon the NIST digit database. For each image in the NIST database, the pixels values were added for zone. These summed zoned pixel features were then split into the various classes and the mean and standard deviation statistics estimated. . . . .	70
4-3 NIST: Fourier transforms of sample digits. . . . .	74
4-4 Gabor: Imaginary ( <i>top</i> ) and real ( <i>bottom</i> ) parts, where $\phi = (0.5, 0.5, 0, 0, 1, 1)^T$ .	77
4-5 Seascape: Distribution of two zoning features. . . . .	80
4-6 Seascape: Change in motor boat class Gabor features with rotation. . . . .	82
4-7 Seascape: Fourier based features showing separability of buoy subclasses. . .	83

4-8 Seascape: Sailboat subclass sail states. . . . . 84

4-9 Seascape: Sailboat subclass designs. . . . . 84

4-10 NIST: Increasing number of intuitive features. . . . . 95

4-11 Seascape: 7-NN results for transform features chosen by Wilks'  $\Lambda$ . . . . . 96

4-12 Seascape: Linear classifier results for transform features chosen by Wilks'  $\Lambda$ . 97

4-13 Seascape: Low Wilks' score indicates good, in this case, Fourier features. . . 97

4-14 NIST: 7-NN classifier results for transform features chosen by Wilks'  $\Lambda$ . . . . 98

4-15 NIST: Linear classifier results for transform features chosen by Wilks'  $\Lambda$ . . . . 99

4-16 The multi-modality problem with individual feature selection. . . . . 99

4-17 Increasing the number of available features. . . . . 100

4-18 Seascape: 7-NN classifier results for geometrical moment features. . . . . 103

4-19 Seascape: The rogues' gallery - objects that were always misclassified. . . . . 107

5-1 Architectural representation of a linear adaptive wavelet (kernel) classifier with one kernel,  $\psi$ , highlighted in bold. Input images are multiplied by a kernel and summed to generate features in the first layer. The second layer acts as a simple linear discriminant. . . . . 113

5-2 Example of the real part of Gabor transform. . . . . 117

5-3 Seascape:  $M = 16$  adaptive Gaussian variance resulting super-kernels. Bright areas represent image locations where the effect on the final classification of an object is biased towards the class of the super-kernel (positive effect), grey areas are where objects do not effect the class decision (nil effect) and black where the effect an object, at that location, is against the super-kernel class decision (negative effect). . . . . 120

5-4 NIST:  $M = 25$  adaptive Gaussian variance resulting super-kernels. Note the images likeness, or partial likeness, to individual digits. . . . . 121

5-5 Seascape: Final centre positions of  $(x_0, y_0)$  parameter vector from the adaptive kernel positioning experiment. Identical kernel starting positions but different splits of the object database. Key:  $\triangle, \square, \diamond, +, x$  represent the results from different splits of the data. . . . . 124

5-6 Seascape: As with the previous Figure but a different starting position. Again, different splits of the object database were used. Key:  $\triangle, \square, \diamond, +, x$  represent the results from different splits of the data. . . . . 125

5-7 NIST: Final centroid positions for 9 kernel model. Key:  $\triangle, \square, \diamond, +, x$  represent the results from different splits of the data. . . . . 126

5-8 Seascape: Kernel centre trajectories problem. . . . . 127

5-9 Examples with and without penalty influence. . . . . 128

5-10 Seascape: Final centre positions of  $(x_0, y_0, \lambda')$  parameter vector using regularised kernel positioning. The same initial starting conditions were used but with different splits of the object database. Key:  $\triangle, \square, \diamond, +, x$  represent the results from different splits of the data. . . . . 129

5-11 Seascape:  $M = 9$  adaptive Gaussian variance and centres resulting super-kernels. Bright spot at top of sailboat kernel will heavily support the case for a sailboat classification if strong object image energy located at this point e.g. a mast. Conversely, this energy will strongly hint against the motor, and more especially, the buoy class. . . . . 130

5-12 NIST:  $M = 15$  adaptive Gaussian variance and centres resulting super-kernels. 132

5-13 Seascape: Three kernels with  $(a, b, x_0, y_0)$  adaptive vector. . . . . 133

5-14 Seascape: Twelve kernels with  $(a, b, x_0, y_0)$  adaptive vector. . . . . 134

5-15 Seascape:  $M = 9$  adaptive 6 parameter Gabor resulting super-kernels. . . . . 136

5-16 NIST:  $M = 15$  adaptive 6 parameter Gabor resulting super-kernels. . . . . 137

5-17 Seascape: Adaptive linear results. . . . . 137

5–18 NIST: Adaptive linear results. . . . . 138

5–19 Seascape: Classification against number of adaptive model parameters. . . . . 141

6–1 Translation, rotation, and scaling. . . . . 144

6–2 Polar plots. . . . . 146

6–3 Trajectory of a transformed pattern,  $k(\gamma)f$  where  $k(0) = 1$ . . . . . 148

6–4 The neocognitron. . . . . 149

6–5 Examples of two different classes of radial polynomials,  $g(\rho)$ . . . . . 155

6–6 The real and imaginary kernels of one Zernike kernel. . . . . 157

6–7 Seascape: Effect on non-RI feature classification by small sensor rotations. . . . . 160

6–8 Seascape: The motor boat, on the left, has been rotated by 80 counterclockwise. 162

6–9 The adaptive complex kernel classifier model. . . . . 163

6–10 Mean squared error (MSE) surface for a simple problem. . . . . 164

6–11 Seascape: Increasing the number of feature kernels. . . . . 165

6–12 Seascape: Increasing the number of complex FMM inputs. . . . . 165

6–13 Seascape: Noise results for the adaptive model. . . . . 166

6–14 Seascape: Final radial polynomials ( $m = 2$ ) for the adaptive model. . . . . 166

6–15 Seascape: 6 radial polynomials ( $m = 4$ ) after 0 and 10,000 iterations. . . . . 168

6–16 Seascape: Adaptive RI linear classification results. . . . . 169

6–17 Seascape: Effect of noise adaptive RI linear model performance. . . . . 170

6–18 Seascape: Combined filter weights for sailboat class. . . . . 171

6–19 Seascape: Tracking of kernel centroids during optimisation and final shape. . . . . 173

7-1 Seascape: Classification versus segmentation quality. . . . . 179

7-2 Seascape: Badly segmented Gaussian feature distribution. . . . . 180

7-3 Seascape: Poorly-segmented object classification. . . . . 183

7-4 Examples of two type of density estimator. . . . . 189

7-5 Seascape: Distribution of novelty values for both non-rogue and clutter data. . 191

7-6 Seascape: Distribution of novelty values for both non-rogue and poorly-segmented data. . . . . 192

7-7 Seascape: Novelty distribution of Fourier features of 3 classes. . . . . 193

7-8 Car: Validation set classification rate, validation set error and training set error during optimisation for the nonlinear 9 kernel, 12 hidden unit adaptive model. 199

A-1 Steepest descent: Problems of oscillation. . . . . 218

A-2 Simplex: Possible steps in two dimensions. . . . . 219

A-3 Conjugate gradient optimisation. . . . . 220

---

# List of Tables

---

2-1	Seascape: Human subject classification results. . . . .	33
2-2	Seascape: Confusion matrix for human classifiers. . . . .	33
3-1	Typical segmentation parameter values. . . . .	41
3-2	Seascape: Segmentation quality of all non-clutter objects. . . . .	46
3-3	Seascape: Object characteristics divided into five levels. . . . .	49
3-4	Seascape: Mean percentage of bounding box area filled by object. . . . .	54
3-5	Seascape: Analysis conclusions. . . . .	60
4-1	Statistical features. . . . .	69
4-2	Low order regular moments. . . . .	72
4-3	Various unitary transforms of a sailboat ( $\alpha = constant$ ). . . . .	75
4-4	Various separability measures. . . . .	81
4-5	Types of classifier implemented. . . . .	87
4-6	Seascape: Separability measures ordered by smallest estimated error. . . . .	89
4-7	Seascape: Classification results. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests. The values in bold represent highest mean classification for a type of classifier and the values underlined the highest mean classification for a type of feature. . . . .	90

4-8 Seascape: Classification results (*continued*). . . . . 91

4-9 NIST: Classification results. Each score is the mean percentage classification over 10 different samples each consisting of 800 test vectors. The value in parentheses is the standard deviation over the 10 tests. The values in bold represent highest mean classification for a type of classifier and the values underlined the highest mean classification for a type of feature. . . . . 92

4-10 Seascape: Gabor features chosen using branch and bound algorithm. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. . . . . 101

4-11 Seascape: Confusion matrix for classifiers based on height and width features. 104

4-12 Seascape: Confusion matrix for classifiers based on 16 Gaussian features. . . 105

4-13 Seascape: Confusion matrix for classifiers based on Wilks'-based Fourier features. . . . . 105

4-14 NIST: Confusion matrix for 7-NN classifier, 16 Gabor features (78.75% correct). 106

4-15 NIST: Confusion matrix for 7-NN classifier, 16 zone features (92.0% correct). 106

5-1 Examples of feature extraction kernels . . . . . 111

5-2 Seascape: Adapting Gaussian variance parameters. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests. The value T represents the total number of parameters in the model. . . . . 119

5-3 NIST: Adapting Gaussian variance parameters. Each score is the mean percentage classification over 10 different samples each consisting of 800 test vectors. The value in parentheses is the standard deviation over the 10 tests. The value T represents the total number of parameters in the model. . . . . 119

5-4 Seascape: Adapting Gaussian kernel positions. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests. The value T represents the total number of parameters in the model. . . . . 122

5-5 NIST: Adapting Gaussian kernel positions. Each score is the mean percentage classification over 10 different samples each consisting of 800 test vectors. The value in parentheses is the standard deviation over the 10 tests. The value T represents the total number of parameters in the model. . . . . 123

5-6 Seascape: Adapting Gaussian kernel positions and widths. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests. The value T represents the total number of parameters in the model. . . 130

5-7 NIST: Adapting Gaussian kernel positions and widths. Each score is the mean percentage classification over 10 different samples each consisting of 800 test vectors. The value in parentheses is the standard deviation over the 10 tests. The value T represents the total number of parameters in the model. . . . . 131

5-8 Seascape: Confusion matrices for 6 kernel linear classifiers. . . . . 132

5-9 Seascape: Adapting all 6 Gabor kernel parameters. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests. The value T represents the total number of parameters in the model. . . . . 135

5-10 NIST: Adapting all 6 Gabor kernel parameters. Each score is the mean percentage classification over 10 different samples each consisting of 800 test vectors. The value in parentheses is the standard deviation over the 10 tests. The value T represents the total number of parameters in the model. . . . . 136

5-11 Seascape:  $M = 4$  nonlinear adaptive kernel model classification results. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests. . . . . 139



5–12 Seascape:  $M = 4$  nonlinear adaptive kernel model parameter count. . . . . 140

6–1 Seascape: Non-RI features with a rotated database. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests. . 160

6–2 Seascape: RI features with a rotated database. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests. . . . . 161

6–3 Seascape: 7-NN classifier confusion matrices. . . . . 162

6–4 Seascape: RI features with a rotated database. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests. . . . . 172

7–1 Seascape: Clutter classification using a 3-category linear and 7-NN classifier. . 177

7–2 Seascape: Effect of segmentation quality on MLP training and classification. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests. . . . . 181

7–3 Seascape: Clutter classification using a 4-categories of data. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests. 184

7–4 Seascape: Confusion matrices for the linear adaptive networks with a clutter class. . . . . 185

7–5 Seascape: All badly segmented data classified using 4-categories. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests. . . . . 186

7-6 Seascape: (EX0 IN3) badly segmented data classified using 4-categories. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests. . . . . 187

7-7 Seascape: Confusion matrices for the fixed feature 7-NN classifiers with an (EX3 IN0) class. . . . . 187

7-8 Seascape: Confusion matrices for novelty classifier using Fourier features. . . 194

7-9 Seascape: Confusion matrices for novelty classifier using Fourier features. . . 194

7-10 Car: Class distributions. . . . . 197

7-11 Car: Results for the infrared vehicle data using fixed features. Each score is the mean percentage classification over 10 different samples each consisting of 500 test vectors. The value in parentheses is the standard deviation over the 10 tests. . . . . 197

7-12 Car: Results for the infrared vehicle data using adaptive features. Each score is the mean percentage classification over 10 different samples each consisting of 500 test vectors. The value in parentheses is the standard deviation over the 10 tests. . . . . 198

7-13 Car: Confusion matrices for the linear and nonlinear adaptive classifiers. . . . 198

---

# Chapter 1

## Thesis overview

---

The thesis describes a project to enhance a section of an existing industrial system. A brief, preliminary, discussion outlines the naivety of this first scheme and proposes a new approach combining available, and novel, technology to simplify, and to enhance the performance of the product. To this end, a list of project aims is presented. The approach taken to achieve these new goals is then explained with reference to each chapter, with a note on why each particular chapter is important. Finally, the contributions to knowledge, that the thesis accomplishes, is discussed.

### 1.1 Thesis background

Automatic Target Recognition (ATR) is the automatic detection, isolation and identification of hostile objects in a real-world environment. The two main goals of a mainstream ATR system are, firstly, the detection of all potentially hostile objects in an environment, whilst minimising the number of false detections, and, secondly, the identification of all detected objects.

A required improvement in both of these ATR performance measures in an existing system, employed by British Aerospace Systems and Equipment Ltd. (BASE), led to the proposal of three interesting research topics for the Electrical Engineering department at Edinburgh University, to be carried out in overlapping phases. The objectives of the project phases, dictated by BASE, included

- I: Improved rate of correct identification of detected objects, using still, infrared, images, without greatly decreasing classifier throughput, or increasing module storage requirements.
- II: Improved detection, and extraction of objects, using feedback, for reducing false detection rates.
- III: Improved object classification using both temporal, and three-dimensional object characteristics.

A fourth phase, using classification to aid object tracking, was also planned and together this entire project was labelled, by the Edinburgh group, as the OSTRICH project: Object Segmentation and Tracking using a Real-time Infrared Classification Hypothesis. The OSTRICH system design consists of all the modules required for a fully working ATR system, and was proposed by the Edinburgh group such that existing BASE components could be used, and that any new, improved, modules tested in the OSTRICH system, could be easily transferred back into the BASE system.

This thesis concentrates on Phase I of the OSTRICH project, improving the rate of correct object identification achieved by the existing BASE classifier. The existing system uses a Multi-Layer Perceptron (MLP), neural network, classifier designed with object image data segmented from a database of thermal infrared images. The original proposal for this thesis was to harmonise work already completed on the recognition system at BASE, with research carried out in the Integrated Systems Group (ISG) at Edinburgh University on improved classification through noisy learning in neural network classifiers <sup>1</sup>.

Unfortunately, as will be explained in detail in later chapters, the original classification results obtained by BASE were highly optimistic, due to a highly over-parameterised classification model and inappropriate test image database. The subsequent need for standard image processing techniques, to generate a set of low-dimensional object characteristics, or *features*, invalidated the requirement for research in connection with noisy learning, this area having

---

<sup>1</sup>Applied Research project number 82140761,"Noise in Neural Training: Infrared Image Classification"

been recently investigated [32]. Consequently, a new approach and set of *project* aims, was considered. The objective of Phase I though, the *BASE* aims, remained as set out earlier in this section.

Before examining the aims of the project it is important to discuss why the particular ATR methodology implemented in this thesis was used in preference to other known systems. The ATR methodology used in this thesis, as alluded to previously, is based on segmenting objects from a scene, deriving a set of features and then performing an object classification based on these features. This is a common approach used in many systems but there are other approaches, such as model matching, CORT-X filtering and the use of knowledge-bases and expert systems. [15,96,29,10]. The reasons for choosing the segment-feature-classify approach is listed below.

1. Each of the approaches have their disadvantages. The approach taken in this thesis could potentially be confused by decoys, often because of the lack of range data and contextual information. In other systems, such as model matching, object occlusion can cause loss of symbol data, and clutter may produce false symbols and knowledge-based systems often require basic object detection and shape recognition in order to generate some of its decisions and, thus again, liable to the same problems as the previous approaches.
2. For these very reasons listed in Point 1 these different approaches are sometimes used in parallel, and combined as to integrate the positive attributes of each of the approaches, and minimising the effects of the failures of each individual approach. Therefore, it is quite acceptable to examine, and improve the performance, of just one of these ATR methodologies.
3. There are many existing systems that are in use that employ the segment-feature-classify approach and improvements to the approach are thus immediately beneficial to these systems.
4. The original BASE system was based on this methodology and continuing with the same approach allowed a small manageable project to be contained.

## 1.2 Project aims

The aims of the project are listed below.

1. To highlight problems with the existing BASE ATR object classification module.
2. To design a replacement classification module for the BASE ATR system. The module must be able to work on real-world data generated by the detection, and isolation, BASE ATR stages. Furthermore, the module must not require more storage than the existing BASE system, nor should its throughput be reduced.
3. To provide improved classification, to be compared, not only with the existing BASE system performance, but also with the traditional approaches of improving classification rates. This enhancement is to be done using a single stage classification process, using image input data, yet maintaining a low number of adaptive classifier model parameters.
4. To design a classifier that is adaptive to new environments and applications. The classifier should also be easy to generate and have a minimal number of control settings. Traditional approaches often fail, if performed correctly, in one or more of these characteristics.
5. To analyse the real data provided for the project, and the processes used in generating the classifier inputs, including determining any assumptions that were made in these steps.
6. To incorporate invariance to size, position, or two-dimensional rotations of the object image, into the classification model. This ability to continue correctly classifying objects regardless of particular object deformations is a very important attribute of an ATR system.
7. To identify potential weakness in the new classification module. This identification must include analysis of classification failures, and examination of the effects of failure of any weak assumptions made in the generation of the data. The latter will require the module to be able to detect non-object, rogue, or inaccurately generated, data from previous ATR stages.

## 1.3 Thesis outline

This thesis is a chronological, and systematic, presentation of the work completed in achieving the aims set down in Section 1.2. The initial chapters provide background information to the project, the data used and how it was produced. The thesis then covers the standard methods for automatically processing the data introduced in the previous chapters. Then, a relatively new approach for simplifying these methods is reviewed, applied, and extended to improve both functionality, and performance. This new model, achieving many of the project aims, is then subjected to more realistic data to fully test its capabilities. A more detailed breakdown of all the chapters is now provided.

Chapter 2 introduces the various basic techniques involved in pattern recognition and Automatic Target Recognition (ATR.) Many of these techniques are used in later chapters. An outline of the OSTRICH project is also provided to clarify the needs and functions of an improved classification module, as is a description of the raw data that was provided, and acquired, for the project. The significance of the chapter is that it provides much of the necessary background for understanding why, and how, the project commenced.

The processing of the raw image data, described in the previous chapter, for generating a set of object databases for classification is described in Chapter 3. This chapter is specific to the database provided for the OSTRICH project, and covers object segmentation, analysis and preprocessing. Objects are extracted from their parent images, labelled according to a defined classification tree, analysed to determine specific object attributes useful for discrimination, and finally, preprocessed to remove any unwanted characteristics that generate misleading, or unrepresentative, information. This chapter lists the assumptions that were made when generating the object data, and is significant as it provides the basis on which satisfactory, and realistic, solutions can be reached.

Before any new improved classification model can be examined it is necessary to process the data using techniques currently employed in other systems. The object identification technique, in use in many classification and ATR systems, currently uses a two stage method of feature extraction and selection, followed by feature classification [10,13]. This is applied, as explained in Chapter 4, to the object databases with many popular feature extraction algorithms

and classification models implemented. This chapter is significant as it provides a set of benchmark results against which any new ATR classification module can be compared, and shows where improvements can be made, with reference to specific types of object classification failures.

Chapter 5 introduces the relatively new, single stage approach, for object classification that uses adaptive kernel feature extraction combined with a standard linear discrimination procedure. This is applied to the databases used in Chapter 4. The results are contrasted with those of the previous chapter. The model is then extended to use a nonlinear discrimination procedure in an attempt to improve performance further. This chapter is significant as it shows how to easily generate a classification model that produces good generalisation without the need for the unwanted complexities of the standard approaches.

The model analysed in Chapter 5 lacks invariance to a required set of object deformations. This invariance is very important to a real-world ATR system. Thus, Chapter 6 further extends the work of the previous chapter and attempts two solutions to this invariance problem. One result is shown to be far superior, and is compared with standard methods for achieving invariance. This chapter is significant as it extends the knowledge of the new classification model, as well as satisfying the invariance requirement given in Section 1.2.

Chapter 7 investigates the effect on the standard, and new, classification models when some of the assumptions concerning the creation of the object database are relaxed. This relaxation generates far more realistic data than the more idealised data used up to this point. This chapter examines methods for identifying much of the rogue data that is now passed to the classification module. This is the preliminary work required for the second phase of the OSTRICH project that is being completed in parallel with the phase described in this thesis. Also in Chapter 7, the new classification ATR module is put to the ultimate test of flexibility to real-world environments when it is exposed to a completely new database. This chapter is significant as it tests the integration of the new classification model into the ATR environment.

Chapter 8 summarises, and provides the final conclusions to, the thesis.



## 1.4 Areas of contribution

Publicised classification results derived from real, infrared, image databases, of the kind used in this thesis, are scarce. This is often due to the nature of the sponsoring companies products. Results from databases where there is exceptionally high clutter to object ratios, object variability, poor image quality, and object obscuration are even less frequently reported. This is perhaps due to the disappointing classification rates compared with synthetic database results. This thesis uses such real-world data, and not only applies a relatively new type of classification algorithm to it, but analyses the practical implications of a non-ideal, classifier-data-generation, mechanism.

The combined feature extraction and classification model used is not itself novel. However, as just stated, the application of the model for differentiating between these type of real, and non-ideal, objects is unknown, at the time of writing, to the author. Furthermore, the combined model is extended in this thesis, and provides new information, with respect to three important, and different, aspects.

- The model used is greatly simplified in terms of the feature extraction mechanism, the form of the mother kernel. Many other authors have adopted a multi-parameter, wavelet, kernel, without, it seems, testing the possible usage of a much simpler, single, or dual parameter kernel. This simplification *is* tested, and analysed, in this thesis using the real, non-ideal data.
- The linear discrimination algorithm used in the combined model is replaced, and tested, with a nonlinear classification algorithm, with only an extra layer of processing. This has been suggested but it has never been applied, or results published, using real-world data.
- Invariance to size, position, and two-dimensional rotation, is incorporated and tested in the combined model. Invariance in the new model, as far as is known, has never been attempted with neither synthetic, nor real-world, data.

## **1.5 Review**

This chapter has described the project aims and the format of the following thesis. It has illuminated the areas in which this thesis will uncover fresh results and ideas.

---

## Chapter 2

# Automatic target recognition and the OSTRICH project

---

Automatic target recognition (ATR) systems are designed to detect, isolate, identify and track user-defined objects of interest within a potentially hostile environment. This chapter provides the necessary background for understanding various ATR, and classification, concepts with particular reference to a specific ATR system. This system is shown to have several fundamental flaws, each of which this thesis shall address. Also, a real-world ATR scenario is outlined and two simple experiments demonstrate that this type of recognition, on real infrared data, is definitely not as easy as it may sound!

## 2.1 Recognition

Great White Sharks, contrary to popular belief, have excellent visual acuity. Even so, on occasion, they attack humans. One theory suggests that sharks mistake the human outline with that of a seal. In comparison, the retinal ganglion response of the frog [5], performing gross visual feature extraction, compels the frog to strike at any suspected prey albeit a fly at 10cm or a plane at 1km. The frog wrongly identifies the prey from a lack of depth information and not by shape, like the shark. These examples merely illustrate that lack of complete, or misguided, information can lead to errors in recognition.

These mistakes may seem simplistic but even humans are not immune to errors in visual perception, especially in tasks which tend to habituation or are stressful due to rapidly developing, or hostile, environments. In these situations humans are slow, unreliable, and vulnerable [131].

These problems are often compounded when the focus of attention undergoes geometrical distortions, such as scaling, repositioning and often, more importantly, rotation [110].

There became a requirement to develop automated vision systems that were fast, accurate, expendable, and could cope with common geometrical distortions in high risk military situations, or where accurate, high throughput was required, such as medical imaging for diagnosis [131]. One particular research area that has received a lot of attention, is automatic target recognition.

## 2.2 Automatic target recognition (ATR)

The term "automatic target recognition" was coined in the early 1980's with the development of the LANTIRN<sup>1</sup> system and now represents the specific field of military-based image analysis and machine vision [102]. Since LANTIRN there have been many ATR related projects including Honeywell's PATS project, the SAIRS program, PAIRSTECH, KMBAA, Hughes' SAHTIRN target recognition system, and the ANVIL program [95,94]. Good introductions to the historical aspects of ATR, spanning 35 years of research, can be found in articles by Roth [91], Brown and Swonger [17], Bhanu [10], and two recent special journal issues dedicated to ATR [94,51]. However, to define an ATR system first requires knowledge of the expected ATR environment.

ATR systems typically operate in military environments where there exist many constituent types of data. In this thesis four components are identified:

- **Objects:** These are components of specific interest. They are subdivided into particular categories and influence the action that *must* be taken by the ATR system.
- **Clutter:** The components that have properties, for example heat radiation, similar to all classes of objects, yet are irrelevant, and must be rejected by the system with no action taken.

---

<sup>1</sup>Low Altitude Navigation and Targeting Infrared for Night

- **Background:** The remaining components of no specific interest on which, again, no action is invoked.
- **Noise:** This is the global, disruptive, property of the environment on the other three types and is dependent on sensor quality, climatic and atmospheric conditions.

The modern ATR system arose as a method for distinguishing and processing these type of components, and is defined, in this thesis, as a pipelined processor of multi-sensor sequence data, often including visual and infrared images, to detect, isolate, identify and track objects in a high risk, military, environment. Within this context, the system has to make decisions, offer suggestions, or even perform actions, based on two fundamental objectives:

- **Primary:** To detect all objects, at all times, in a hostile environment whilst maintaining a low clutter detection rate.
  - **Secondary:** To distinguish between object categories as well as between objects and clutter.

The most common approach to satisfying these objectives is to subdivide the ATR system into several distinct, functional modules [10]. Typically these modules, as shown in Figure 2–1, include data preprocessing, object detection and segmentation, feature extraction, and finally classification. This last stage provides outputs to be interpreted, with all other available information, such that an action can be suggested to an operator.

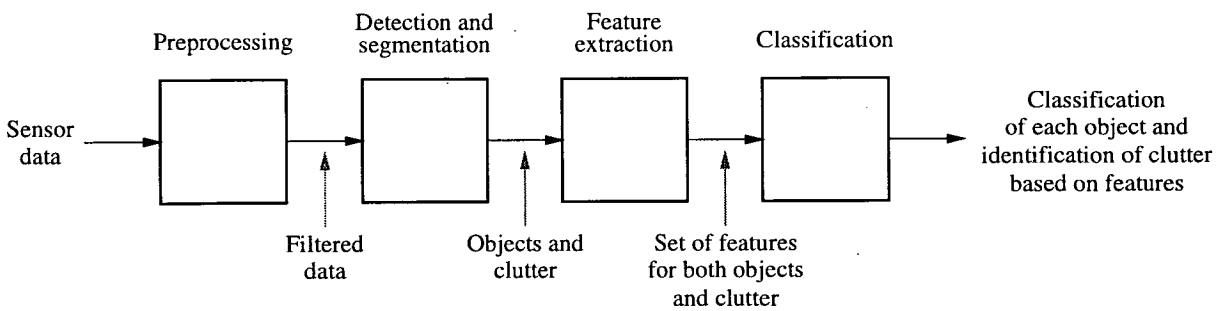


Figure 2–1: A common serial ATR system.

One major drawback of a serial system though is that the output is dependent on the performance of all preceding modules, even the acquisition of the data. An inappropriate set of sensors will partially or even completely fail to register an object’s existence making the problem of detection exceptionally difficult or even impossible. Subsequently, much consideration has been given in the literature to the object-background separability properties of sensors in particular environments [91,76,132].

ATR sensors

There are two main types of vision-based ATR sensor; passive and active. Active sensors transmit electromagnetic energy to illuminate an object surface and receive Doppler shifted backscatter echoes to generate an image. Synthetic aperture radar (SAR) is a good example of an active sensor. SAR is a popular ATR sensor due to the relationship between image magnitude and object range. Unfortunately, active sensors are often susceptible to detection and countermeasures.

Passive sensors, such as forward looking infrared (FLIR) sensors, generate images based on the radiation naturally emitted by an object and so are far less detectable. FLIR sensors typically operate in the 8-12 $\mu m$  region of the electromagnetic spectrum, as shown in Figure 2-2, and react to active thermal signatures. A property common with many man-made objects. This

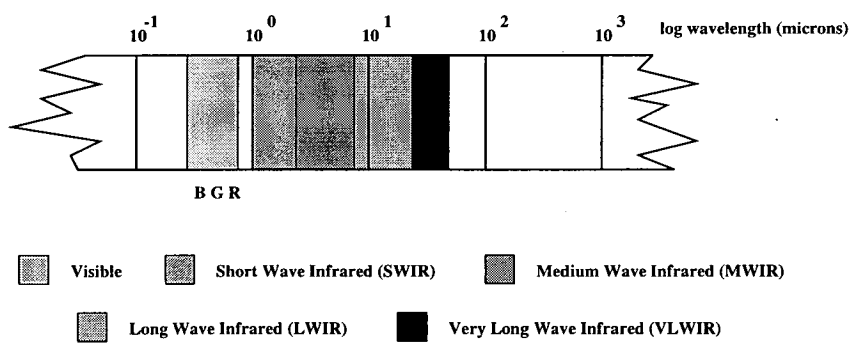


Figure 2-2: A section of the electromagnetic spectrum.

is important in ATR systems as man-made objects are often the components of interest.

There are, of course, considerable drawbacks to FLIR, as given in Figure 2-3, but their

proven technology still make it popular in modern ATR systems. A description of the history, construction and operating environments of thermal passive sensors is given by Norton [76].

The choice of ATR sensor is also problem specific. Requirements may be dictated by a required field of view (FOV), signal-to-noise ratio (SNR), range, atmospheric conditions, day or night operation, possible countermeasures and many other effects. These are outlined in Figure 2–3. One, expensive, solution is to combine different types of sensor.

Roth comments that with multi-sensor fusion *"The utilisation of multiple sensors to acquire data for target detection and recognition is a major consideration in significantly improving ATR performance"* [91]. This improved sensor array is not only restricted to image and range data. Absolute co-ordinate information, external beacons, sensor orientation and climatic information, such as temperature, may also be beneficial to an ATR system.

## **Preprocessing**

Preprocessing is required to enhance signals before any further operations are performed. Image enhancement is designed to improve object contrast and reduce noise, as well as to control image focus, gain and bias. Suitable techniques include median filtering, unsharp masking, and histogram equalisation [10].

## **Detection and segmentation**

Object detection and segmentation determines the locality of a potential object within a region of interest (ROI) of an image and extracts it from the background as accurately as possible. This is the primary objective of an ATR system as stated earlier. An undetected object can not be classified.

SENSOR	ADVANTAGES	DISADVANTAGES	NOTES
Forward Looking InfraRed (FLIR)  (8-12 micron)	High target/background contrast Day/night operation All weather operation Wide FOV No shadows Mature technology Penetrates fog/smoke/dust/haze Very effective detector of heat generating objects	Target signature variability Undesirable degrees of freedom High false alarm rate from background clutter Range uncertainty High in irrelevant information Little information in absolute magnitudes	Medium resolution Passive
Synthetic Aperture Radar (SAR)	Day/night operation All weather operation High target/background contrast Wide FOV Deeper penetration than FLIR through vegetation Less sensitive to target operating conditions and environment Amplitude contains target size data Penetrates fog/smoke/dust/haze	Long dwell time on target Precise tracking and stabilisation Complex technology Power requirements Image speckle	Large dynamic range (typically 5 orders of magnitude) Based on reflectivity differences Medium/high resolution Speckle due to coherent processing (characteristic of multiplicative noise) Size invariant of distance to target
Inverse Synthetic Aperture Radar (ISAR)	As SAR	As SAR	Sensor stationary whilst target moves
Millimetre Wave Radar (MMWR) (3.2mm)	Penetrates fog/smoke/dust/haze Good for close range (large FOV) Day/night operation All weather operation	High false alarm rate from background clutter Target signature variability	
Visible Electrooptical	Light weight Inexpensive High resolution Reliable	Low target/background contrast No all weather capability No all day capability	
Other			Sonar Acoustic time series Multispectral Multisensor Laser Radar (LADAR)

Figure 2-3: A comparison of modern ATR sensors.



## Feature extraction

A set of features are then derived from the localised sensor data, such that the detected objects can be classified. These features must be *discriminative between* and not *representative of* an object. For example, to distinguish between a triangle and a rectangle only requires knowledge of the number of vertices. Other information, such as size, colour, angles, which would be required to reconstruct each object are superfluous. Feature extraction reduces the degeneracy that exists in the sensor data. Furthermore, a small optimal set of features will reduce the computational load of the actual classification whereas a poor set of features may require a highly complex and highly parameterised discriminant.

## Classification

In this thesis, the term classification will be used with reference to the categorisation of an unlabelled object based on a set of previously labelled features. In the neural network literature this is often called *supervised* classification, and in the statistical literature as discriminant analysis [98]. Discrimination is the process of dividing the space spanned by the labelled features into regions such that classification can be performed. The more complex the feature space the harder the discrimination, often producing degraded classification.

One of the main project aims, as set down in Chapter 1, was to improve the original classification ATR module. Thus, it is necessary to understand more of the fundamentals of object classification, before commenting on the original modules performance.

## 2.3 Object classification

At the centre of a typical ATR system are the feature extraction and classification stages. These units are required to identify each detected object based on a fixed set of patterns or features that are derived from each object. This identification is often based on a set of  $N$   $M$ -dimensional feature vectors,  $\mathbf{d} \in \mathbf{D} = \{\mathbf{d}_1, \dots, \mathbf{d}_n, \dots, \mathbf{d}_N\}$ , previously labelled with one of  $C$  classification classes,  $c(\mathbf{d}) \in \Omega$  where  $\Omega = \{\omega_1, \dots, \omega_c, \dots, \omega_C\}$  [30,73,56,40,52,28]. Each of these classes is assumed to have some form of class dependent probability distribution,  $p(\mathbf{d} \mid \omega_i)$  and an  $a$

*priori* probability of occurrence <sup>2</sup>,  $P(\omega_i)$ . The latter is usually estimated from the occurrence of a class in the feature database such as  $N_i/N$  where  $N_i$  is the number of occurrences of a feature vector labelled  $\omega_i$  in the database  $\mathbf{D}$ . If the actual class conditional probability density functions,  $p(\mathbf{d} | \omega_i)$ , can also be sufficiently modelled then the *a posteriori* probability of any object,  $P(\omega_i | \mathbf{d})$ , can be determined from the feature vector using Bayes Theorem which states

$$P(\omega_i | \mathbf{d}) = \frac{p(\mathbf{d} | \omega_i)P(\omega_i)}{p(\mathbf{d})}. \quad (2.1)$$

The value of  $P(\omega_i | \mathbf{d})$  gives the probability that an object belongs to class  $\omega_i$  given a feature vector,  $\mathbf{d}$ . The probability of misclassification is minimised by classifying  $\mathbf{d}$  as of class  $\omega_i$  if  $P(\omega_i | \mathbf{x}) > P(\omega_j | \mathbf{d}) \forall i \neq j$  [13,52,30]. However, the form of the class conditional probability density functions are often unknown. In this case, the task is often reformulated to estimate  $C$  discriminant functions,  $z_k(\mathbf{d})$ , such that  $z_i(\mathbf{d}) > z_j(\mathbf{d}) \forall \mathbf{d} \in \mathbf{D}$  for  $i \neq j$  given that  $c(\mathbf{d}) = \omega_i$ . If  $z_k$  is set equal to  $P(\omega_k | \mathbf{x})$  then the classification decision is based on a requirement of minimising the probability of misclassifying a new pattern. However, this *1-of-C* decision criterion is sometimes not the most appropriate. For example, in ATR there are situations where there are far more serious consequences of misclassifying an object as a non-target than as a target. Although this concept, known as *Bayes risk* [30], is very important to ATR the simple *1-of-C* scheme is suffice at this stage of the investigation of adaptive feature extraction classifiers. More complicated criteria that incorporate risk, for example, can easily be tested at a later stage.

Various statistical models that have the ability to generate approximations of the  $z_k$  functions shall be discussed in the next section. They are divided into two broad categories; parametric and non-parametric. The next sections shall discuss three types of classifiers from both these groups. The critical issue in developing all of these models though is *generalisation*.

Generalisation is a measure of a models ability to classify correctly previously unseen features [90]. Poor generalisation is sometimes attributed to the discriminant functions having too much flexibility and learning the labelled data used to generate the model and not learning the process that generated the data; the underlying probability distributions. Other reasons

---

<sup>2</sup>Notation: In this thesis  $p()$  will be used to define a probability density function and  $P()$  to define a probability [13].

for poor generalisation, and a method for estimating generalisation, is given in the section on Multi-layer perceptrons.

### 2.3.1 Linear classifiers

In particular problems, such as when  $p(\mathbf{d} \mid \omega_i)$  are normally distributed with identical covariance matrices, the Bayes decision boundaries are linear. In this case a linear discriminant is required. These are implemented as single layer networks of the form

$$z_k(\mathbf{d}; \theta) = w_{0k} + \mathbf{w}_k^T \mathbf{d}, \quad (2.2)$$

where  $\theta$  is a vector containing model parameters,  $\mathbf{w}_k$  and,  $w_{0k}$ . These are sometimes known in the literature as *weights* and *biases*.

The parameter vector estimate,  $\hat{\theta}$ , is chosen as to minimise a suitable error criterion,  $E(\theta)$ . A commonly used error function is the *sum-of-squares error* (SSE), and is defined as

$$SSE(\theta) = \sum_{n=1}^N \sum_{k=1}^C \{z_k(\mathbf{d}_n; \theta) - t_{kn}\}^2 \quad (2.3)$$

where  $t_{kn}$  is the target value for an observation,  $n$ , with a *1-of-C* output encoding scheme where

$$\begin{aligned} t_{kn} &= 1 \quad \text{for } k = \omega_n \\ &= 0 \quad \text{otherwise.} \end{aligned} \quad (2.4)$$

The least squares (LS) estimation method that minimises  $E(\theta)$  is found by either iterative techniques such as steepest descent or conjugate gradients (see Appendix A), or more directly using a pseudo-inverse [13].

Linear discriminants are parametric and are a subset of much larger class of functions known as generalised linear discriminants (GLD's), which use predefined functional transformations of input features [28,40,30]. The GLD is defined as

$$z_k(\mathbf{d}; \theta) = w_{0k} + \sum_{j=1}^M w_{jk} \varphi_j(\mathbf{d}), \quad (2.5)$$

where  $\varphi$  is known as the basis function. An example of the GLD is the quadratic classifier which uses a second order polynomial discrimination boundary. Quadratic boundaries are Bayes optimal for normally-distributed data when the the covariance matrices differ. An example of this is given in Figure 2–4.

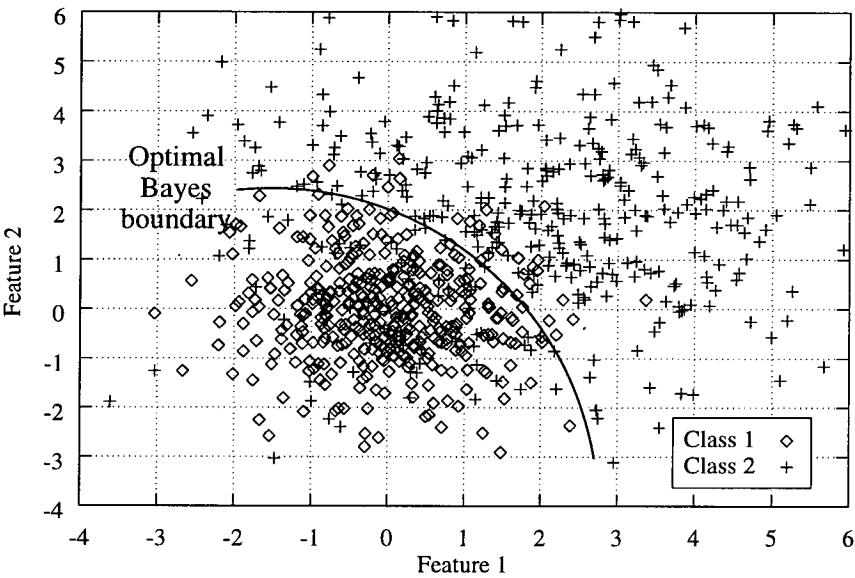


Figure 2–4: Optimal quadratic decision boundary for a two class problem.

2.3.2 K-nearest neighbour classifiers

The  $k$ -nearest neighbour algorithm examines the nearest  $k$  labelled samples, according to a suitable distance metric, for example Euclidean, from an unlabelled point in feature space [24, 52]. If  $k_m$  of the  $k$  samples are of class  $m$  and  $k_m = \max\{k_i\}$  for  $i = 1, \dots, C$  then this point is classified as belonging to class  $\omega_m$ . Alternatively, this can be considered as measuring  $k_i$  of the  $k$  samples in a hyperspherical volume of feature space,  $V$ , where either  $V$  or  $k$  can be adjusted to vary the amount of smoothing applied to what is effectively a piecewise linear classifier. Furthermore, the  $k$ -NN estimator of  $p(d \mid \omega_i)$ , for class  $\omega_m$ , is defined as  $k_m/N_m V$ . However, the  $k$ -NN estimator can *not* be treated as a density function because the integral of the estimator over the feature space does not sum to unity.

The  $k$ -NN algorithm is purely a nonparametric technique and, consequently, has the usual disadvantages of a large memory requirement and the time-consuming need to re-examine every

point in the stored database for each object classification. These two problems make the  $k$ -NN classifier impractical for this ATR system, even though there have been many improvements, such as pruning techniques, in order to reduce the effects [52]. However, it is an excellent method for quickly testing a set of potential features.

2.3.3 Multi-layer perceptron classifiers

It is unimportant to dwell on the historical issues of multi-layer perceptrons (MLP's) and other types of artificial neural networks as it is suffice to say that an MLP is purely *"one of a class of flexible non-linear regression methods which can be used to classify via regression"* [89]. They are a method of parameterising a fairly broad set of non-linear discriminant functions and are in fact are *universal approximators* in that given sufficient complexity and data they can approximate virtually any function [135]. For background information on MLP's, and neural networks in general, there exists an extensive literature [13,68,92,8,54,100]. A more comprehensive viewpoint of neural networks, especially MLP's, from a statistical perspective can be found in [98,89,90,22].

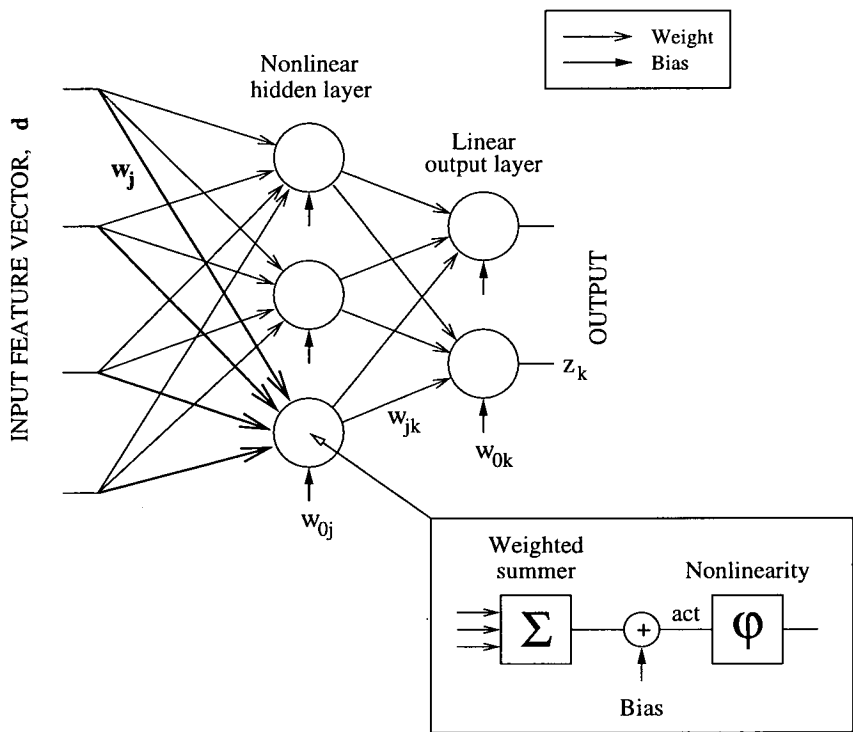


Figure 2-5: Architectural diagram of an MLP.

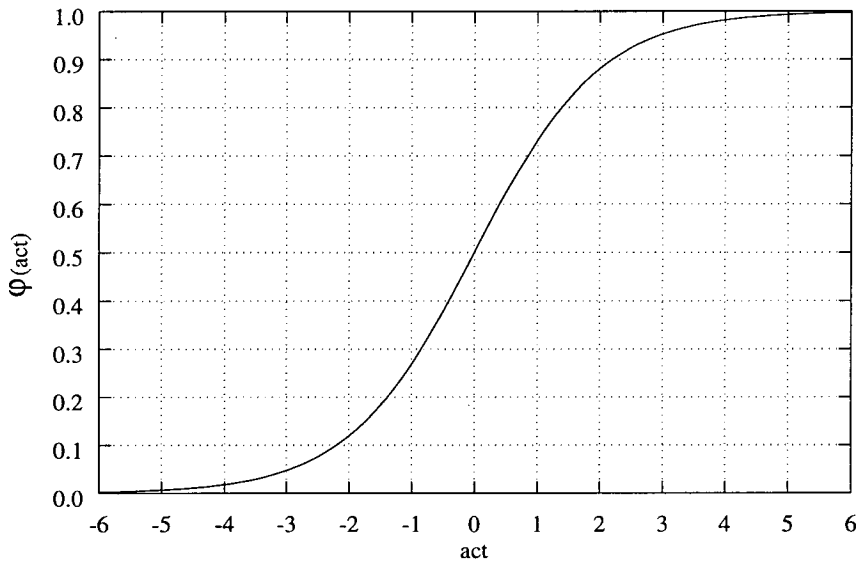
Figure 2–5 shows the architecture of the standard MLP. They comprise, typically, of three layers, known as the input, hidden and output layers, and are defined by the equation

$$z_k(\mathbf{d}; \theta) = w_{0k} + \sum_j^H w_{jk} \phi(w_{0j} + \mathbf{w}_j^T \mathbf{d}) \tag{2.6}$$

where the set of discriminant functions,  $z_d$ , are characterised by the parameter vector,  $\theta$ , which is comprised of all the weights and biases in the network. The hidden layer non-linearity,  $\phi$ , is usually the logistic (sigmoid) function, given in Equation 2.7 and plotted in Figure 2–6.

$$\phi(z) = 1/(1 + e^{-z}), \tag{2.7}$$

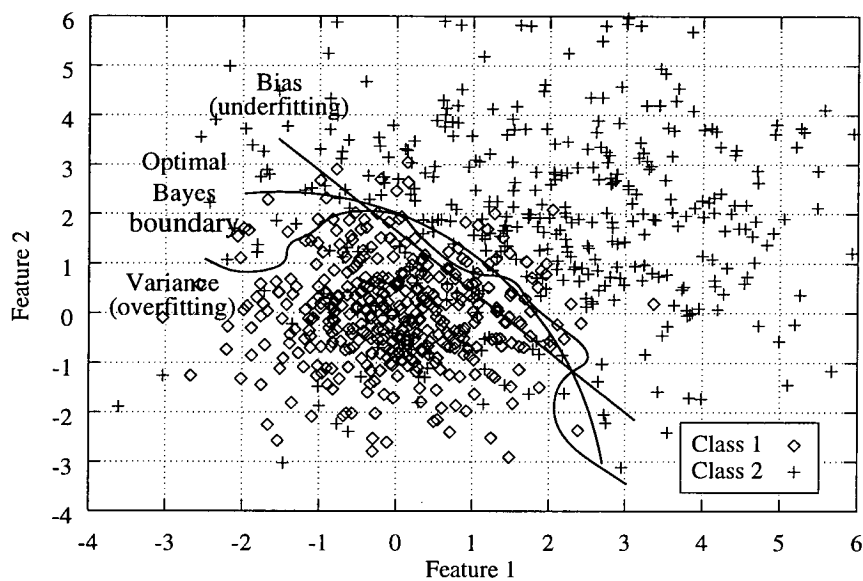
As with the linear classifier, the parameter vector contains the, hopefully small number of, adjustable weights and biases for the model. When estimating these MLP model parameters it is important now to consider both the error function but also the amount of flexibility allowable in the model. There have been many approaches taken to this, data-dependent, problem of



**Figure 2–6:** The logistic function commonly used as the MLP non-linearity,  $\phi$ .

model complexity [13,90]. Too little flexibility causes model *bias* but too much results in *variance* leading to over-fitting of the data and subsequently poor generalisation. An example of this bias-variance dilemma, known as Occam’s Razor, is to attempt to fit a sampled quadratic function with a linear model (bias) or a polynomial with degree greater than two (variance.)

Another example is given in Figure 2–7. The linear fit has not enough flexibility to match the Bayes optimal boundary, which the highly flexible non-linear attempt has over-fitted, resulting in a high classification rate for the training data but poor generalisation. Two ways used to



**Figure 2–7:** Too little, and too much, flexibility resulting in model bias and variance.

control model complexity in this thesis are varying the number of hidden nodes,  $H$ , using *early stopping* to halt optimisation [90] and the regularisation technique known as weight decay [64].

To measure generalisation data is randomly split into three separate sets of data; training, validation, and testing. In this thesis a split ratio of 2:1:1 is used. Model parameters are estimated using the training data. The validation data is used to determine when to stop the optimisation process. The method of early stopping halts the optimisation when the validation set error begins to increase, suggesting over-fitting. Usually though optimisation is performed twice, optimisation stopping when the minimum validation error of the first run is achieved. Generalisation is the classification rate achieved by the model on the independent test data set. Experiments are often repeated several times with different random splits of the data.

Weight decay, which is equivalent to ridge regression, as shown in Equation 2.8, adds a term to the error term,  $E$ , such as to penalise large weights in the model and uses a variable,  $\lambda$ , to control the amount of regularisation. This has the effect of constraining the hidden node inputs to operate in the more linear region of the sigmoid non-linearity thus controlling the

effective model complexity.

$$E'(\theta) = E(\theta) + \frac{\lambda}{2} \sum_i \theta_i^2 \quad (2.8)$$

Earlier the LS estimation method was used for determining parameter estimates for the linear classifiers. The choice of LS is attractive as the derivatives of  $E$ , with respect to the weights and biases, can be backpropagated from output to input by use of the chain rule. This can be extended to the matrix of second derivatives, known as the error Hessian,  $H$ . Both the Hessian and the first derivatives can be used in the iterative optimisation techniques used to minimise  $E$  for the MLP (see Appendix A). Unfortunately, there are drawbacks to LS. These include the fact that LS is not particularly suited to fitting a function with target values of 0 or 1 and is also not particularly robust to outliers in the data [89]. Most importantly though is that the goal of classification is to minimise the misclassification rate and not the SSE error. However, for reasons of simplicity, ease-of-use and the fact that historically it has been shown to work well, LS will be used in this thesis.

The MLP is flexible, adaptive, and requires significantly less storage than its  $k$ -NN non-parametric rival. The decision boundaries are continuous and non-linear, though it is difficult to encapsulate classes in feature space. Finally, its simple, parallel, structure allows for high classification throughput in an easily implementable format.

### 2.3.4 Other classifiers

Two other supervised classifiers used in this thesis include radial basis functions (RBF's) and multivariate adaptive regression splines (MARS).

Radial basis function neural networks approximate functions using linear combinations of non-linear basis functions,  $\varphi$ , centred in feature space [16]. They are identical to the GLD's discussed previously in Equation 2.5 with the exception that the nonlinear function,  $\varphi$ , is typically a Gaussian basis

$$\varphi_j(\mathbf{d}) = \exp \left\{ -\frac{\|\mathbf{d} - \mathbf{d}_{0j}\|^2}{2\sigma_j^2} \right\} \quad (2.9)$$

where  $\mathbf{d}_{0j}$  are basis centres in feature space, and  $\sigma_j$  the width, or coverage, of the basis. These extra parameters can be included in the model parameter vector. In this thesis a fully supervised



approach is taken to estimating all the parameters in the model at the same time, as opposed to the two stage unsupervised approach [13]. This is purely for reasons of simplicity. The advantage of the RBF network is that they can easily form closed decision boundaries.

The other classifier used in this thesis are the MARS classifiers [37]. It is a popular statistical classification model and is defined, like the RBF, with a nonlinear basis, in this case

$$\varphi_j(\mathbf{d}) = \prod_{p=1}^{P_j} \phi_{pj}(\mathbf{d}) \quad (2.10)$$

where the *degree* is the largest value of  $P_j$  and  $\phi$  are, in this thesis, piecewise cubic splines, as suggested by Friedman, for smoothing decision boundaries.

The classifiers considered, and implemented, in this thesis are but a few of the possible models available and were mainly chosen due to their ease-of-use, implementability, and current popularity in the research literature. Other well known classifiers include correlators, projection pursuit regression, logistic discriminants, classification trees, piecewise linear, and many unsupervised techniques. There are also many alternative approaches to estimating model parameters such as MacKay's application of Bayesian inference techniques to neural networks [69,74]. However, there was insufficient time in the project to examine all these types of classifiers and estimation methods.

## 2.4 The OSTRICH ATR system

Figure 2–8 illustrates the ATR model proposed for the Edinburgh OSTRICH (Object Segmentation and Tracking using a Real-time Infrared Classification Hypothesis) project. This model allowed for the use of the existing modules available from BASE and provided scope for future work. The model is simplistic in comparison to the BASE system but allows for testing of new modules, which can then be migrated to the BASE model. The modules that have been constructed to this date, or are currently under development, are shaded in the Figure.

Phase I of the OSTRICH project involved improved feature extraction and classification of large, infrared, objects and clutter,  $O_n^m$ . This is the work described in this thesis. Phase II seeks to improve object segmentation using a resegmentation technique involving localised classification feedback [85,86]. Phase III originally considered temporal classification issues but is now concentrating on the three-dimensional aspects of objects for classification purposes [23].

### 2.4.1 System overview

Each infrared image,  $I_n$ , is enhanced using a separable median filter. This reduces speckle yet preserves object edges. Segmentation of these images then generates both small,  $O_n^s$ , and large objects,  $O_n^m$ , as well as positional,  $X_n$ , and object range data,  $D_n$ . Range is extracted, for example, from a range image,  $R_n$ . The objects of small pixel size, determined by a suitable threshold, can not be classified by shape and must use high level information,  $Q_n$ . This is a subset of the contextual data,  $S_n$ , derived from knowledge bases,  $K_n$ , co-ordinate and climatic data,  $C_n$ , and reference points such as horizons and beacons,  $B_n$ .

The classification of all the objects and clutter provide both probabilities of correct classification,  $P_n$ , and also measures of novelty,  $N_n$ . Phase III will also hopefully determine object poise,  $A_n$ .

A parallel process tracks the objects over series of frames using positional data and previous classification results. Tracking is an important issue in ATR. Significant information can often

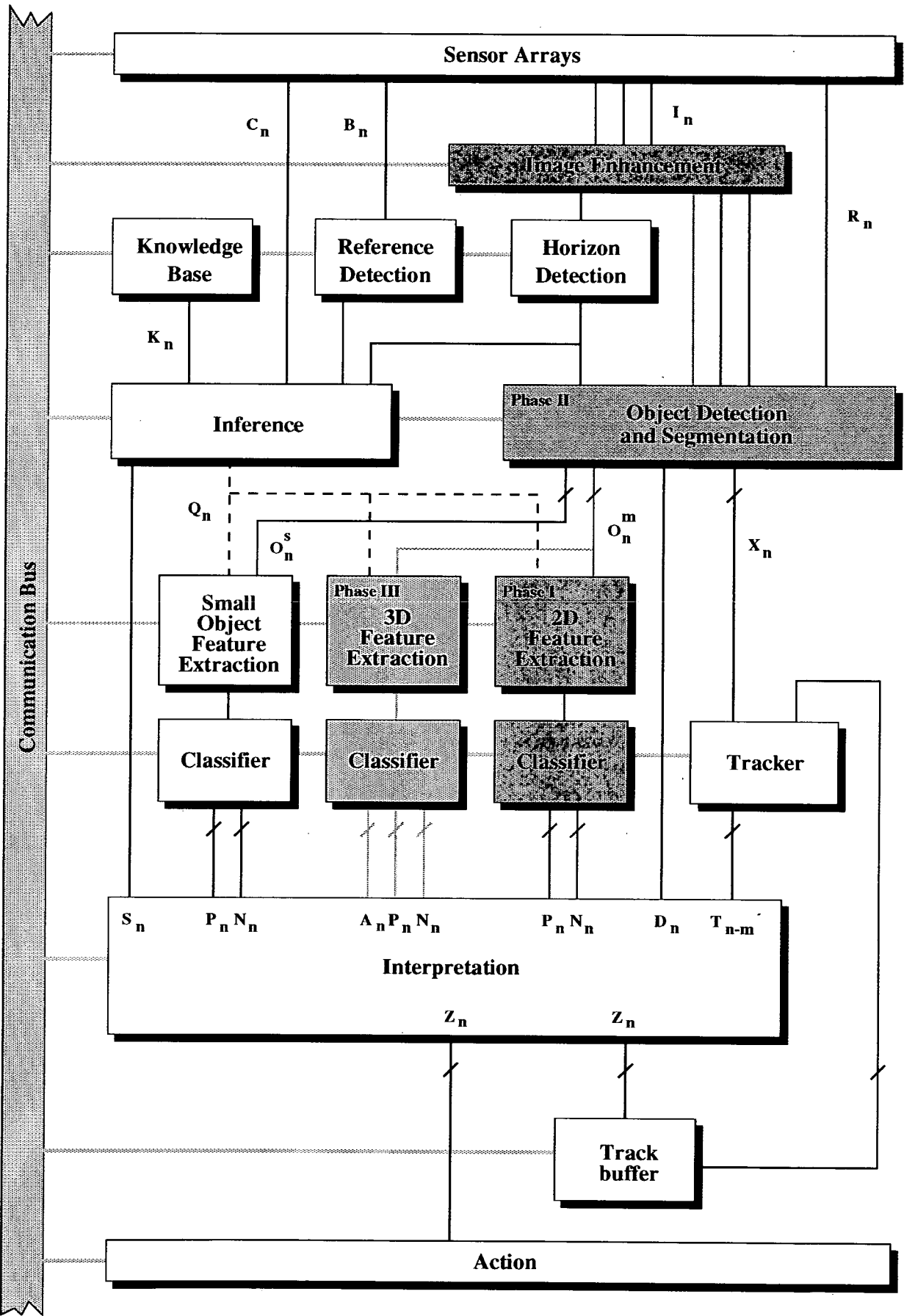


Figure 2-8: System overview at  $n^{th}$  time step

be derived by measuring an object's position ( $x_n$ ), velocity ( $\dot{x}_n$ ), acceleration ( $\ddot{x}_n$ ), and trajectory. Reisher provides a summary of object trackers [87]. Information of this type is sometimes enough to identify an object.

All information is collated by the interpretation module. This could be based on an AI framework, such as an expert system, and combines the contextual, temporal, positional, and classification results to form an overall vector of possible actions for each object,  $Z_n$ . The system may decide to dismiss the object classifications as irrational with the other information dominating.

However, the principal module considered in this thesis, covering Phase I of the project, was the classification unit, and how classification performance could be improved.

## 2.4.2 Original classification module

The original BASE neural classifier is a non-linear, three layer structured MLP with various object images passed directly to the neural network inputs from the segmentation stage. This implied that no separate feature extraction stage was required as the features were derived directly from the actual objects. This classifier had a high throughput, was reasonably simple to implement, and produced excellent classification results with real infrared object data [50, 49,48]. So, why was there a requirement to improve the current module?

## 2.4.3 Problems with the original module

The neural classifier BASE implemented, though appearing to provide excellent classification rates, was flawed in several respects:

1. There was no consideration that the discrimination boundary was, perhaps, linear. The non-linear MLP classifier would overfit the correct solution.
2. The segmentation and detection stages were assumed ideal and were manually adjusted to generate unrealistically well-segmented data.

3. The model was underdetermined. The input layer typically consisted of a 16x16 pixel array, with between 16 and 512 hidden layer units and 5 outputs. With one particular database a model was used that required a total of 133,636 adaptive model parameters to be estimated. These parameters were estimated with typically 2000 labelled samples! Even with the self-correlation that exists within the images, which may partially alleviate the underdeterminedness, the network was acting as a storage element and not generalising to the problem.
4. Due to the lack of data the labelled samples were randomly split into only two separate sets using a uniformly distributed source. The first set was used to derive the model parameters and the second to validate the model during optimisation. The latter was used to determine when optimisation was complete but also used to measure classification performance. The use of no independent test set meant the results were biased. They were biased even further because the results were only based on one random split of the data. This indicates poor generalisation.
5. The labelled data was derived from continuous image sequences. This meant that objects in one frame were highly probable to have a corresponding twin object in the preceding frames. When splitting the data there was then a high likelihood that objects would be split across the sets. Once again this produces biased results, again poor generalisation.
6. There was only rudimentary scaling invariance built into the classifier which meant when objects rotated, or existed any position slightly away from the norm, the classifier failed to correctly identify the object.

#### **2.4.4 A solution**

The previous section outlined several problems with the original BASE classifier. This thesis attempts to address these problems by designing a new ATR classifier using both fixed and adaptive feature extraction techniques to significantly reduce the number of adaptive model parameters, maintain good classification rates, incorporate invariance, whilst still maintaining the attractive MLP architecture.

## 2.5 Databases available

This section describes the data available for testing the standard algorithms, discussed in this chapter, and any new classification module proposed. The data includes:

- Simple test data
- NIST digit database
- Forward-looking infrared images
  - FLIR seascape imagery
  - FLIR land-based imagery

### 2.5.1 Simple test data

These databases include the Fisher iris data and multivariate Gaussian. The Fisher iris data contains 150 examples of various features, such as sepal length, of three different varieties of the iris plant; Iris Setosa, Iris Versicolour and Iris Virginica [34]. This is a classic database used to test discrimination. Another simple data set with a model for the underlying distribution is based on the multivariate Gaussian. A method for generating M-dimensional Gaussian distributed class conditional data, via Equation 2.11, can be found in [83],

$$p(\mathbf{d} \mid \omega_m) = \frac{1}{N_m} \sum_{i=1}^{N_m} \frac{1}{(2\pi)^{(M/2)} |\Sigma_m|^{1/2}} \exp \left\{ -0.5(\mathbf{d}_{0m} - \mathbf{d}_i)^T \Sigma_m^{-1} (\mathbf{d}_{0m} - \mathbf{d}_i) \right\} \quad (2.11)$$

where  $\mathbf{d}$  is the data vector,  $\omega_m$  the required class,  $N_m$  the number of examples in the class,  $\mathbf{d}_{0m}$  the data class mean, and finally,  $\Sigma_m$  the class covariance matrix.

### 2.5.2 NIST digit database

The National Institute of Standards and Technology (NIST) database, *fl3*, is comprised of 3,471 examples of ten handprinted digits from 49 different writers [136]. The samples were collected from a series of handprinting sample forms by the *U.S. Bureau of the Census* with geographical

sampling according to population density within the United States. The forms were scanned at 300 pixels per inch and stored at 8 pixels per byte. Each character on the form was segmented and spatially normalised to a size of 32x32 pixels. A set of examples from the NIST database is shown in Figure 2–9.

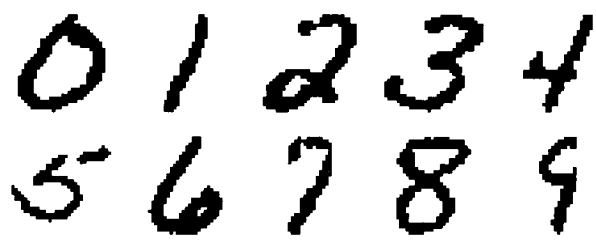


Figure 2–9: NIST: Examples from the fl3 database.

The NIST database was chosen deliberately as it was very different from the BASE data and provided a different application on which to test the flexibility of a new classifier.

2.5.3 Forward-looking infrared (FLIR) images

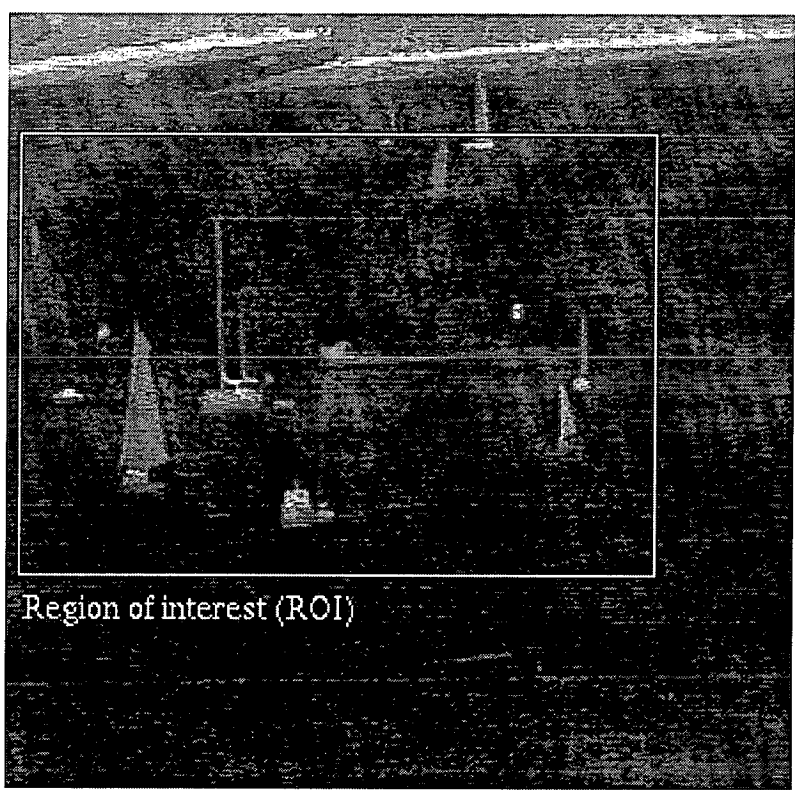
Two sets of real, infrared, image databases were provided by BASE to test the improved ATR classification module. However, only the primary database was readily available to the Edinburgh group, and this was the database used for the majority of the experiments. The second database was used to test the adaptability of the system to new environments. Both were captured with a military thermal sensor.

The thermal sensor used was a class II Thermal Imaging Common Module (TICM II). The TICM is a mechanically scanning, infrared camera operating in the 8–12μm region of the electromagnetic spectrum. Using the camera it was possible to capture 512x512 pixel, 8 bit resolution, image frames consisting of two interlaced field signals. Unfortunately, the interlacing of these fields was not correctly synchronised with this camera but this was solved by considering only single field data, which effectively halved the height of the images.

Another problem was that when objects left or entered the FOV the TICM compensated by altering the thermal window, the linear operating region of the camera, to maintain a constant signal energy in the image. This causes the apparent heat of objects remaining in the scene to

change. Conversely, fixing the position of the thermal window led to object saturation at the extremities.

One particular set of 608 TICM infrared images were extracted from a total of 7 hours video footage of various coastal locations around Falmouth, S.W. England [114]. The images include a variety of sea-faring crafts taken from a constant depression angle and at many different perspectives. The craft were easily detected due to their internal heat sources, and the hot summer weather. The 608 scenes were chosen to minimise the probability of object repeatability, and an example of one of the images is shown in Figure 2–10.



**Figure 2–10:** Seascape: Typical infrared database scene.

An example of the required output of the system to be developed is shown in Figure 2–11. In this example different types of class are labelled with different colours by an automatic classification system. The sail boats are indicated by the colour green, the motor boats blue and the buoys red. The system fails in two particular cases and these are denoted by grey colouring<sup>3</sup>.

<sup>3</sup>This is an actual response from the OSTRICH system.





**Figure 2–11:** Seascape: Example of the required classifier response.

## 2.6 A nontrivial problem

From a human perspective identifying these type of objects contained in the seascape database is an apparently simple task. There is an abundance of information immediately available contained in the scene, as well as, in the actual object. Figure 2–12 though demonstrates how adding different levels of information can significantly ease recognition and reduce the probability of misclassification. In Figure 2–12(a) there exists no scaling, rotational, positional or greyscale information. Adding greyscale data and correcting object orientation, as shown in Figure 2–12(b), is an improvement as information has been added but the object is still difficult to identify. This is the type of object data that will be presented to the classifier. However, Figures 2–12(c) and 2–12(d) demonstrate the benefit of extra contextual information, such as the surrounding objects, the object range, and previous classifications.

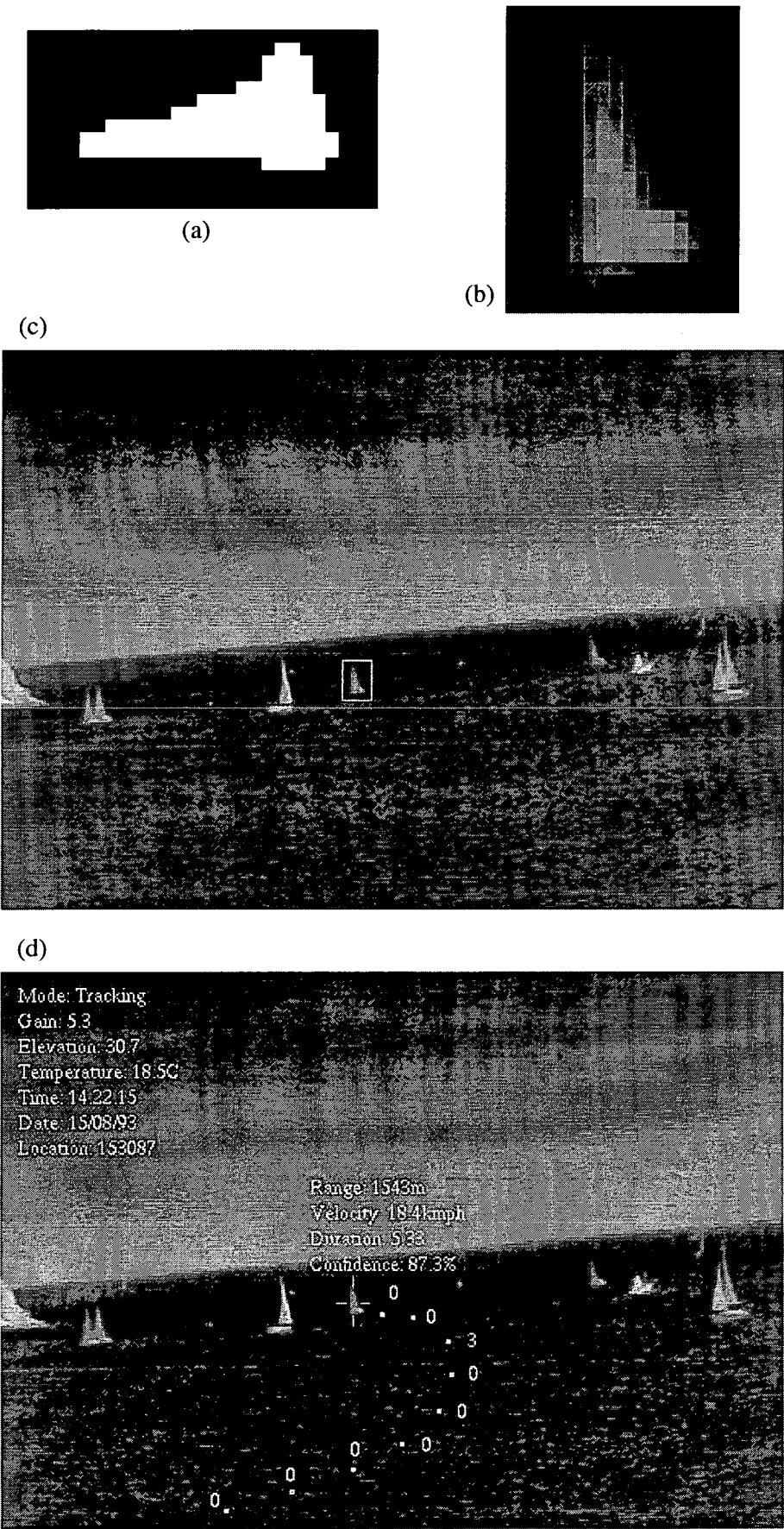


Figure 2-12: Seascape: Different types of information

As a further demonstration of the nontrivial nature of the problem a set of experiments was performed on eight people, one of whom was deemed an expert at classifying objects extracted from the sailboat image database. The other subjects had previous exposure to sample frames from the data and their own mental images of typical seascape objects. In experiment A subjects were each shown 100 random selections from the object database, an example is given in Figure 2–12(b), and asked to classify each object as sailboat, motor boat, buoy or 'anything else'. In experiment B subjects were shown 100 randomly selected objects situated in their original frame and again asked to classify the objects. The results are given in Table 2–1 and sample confusion matrices for experiments A and B are given in Table 2–2.

Subject	Experiment A	Experiment B	Improvement
	(%)	(%)	(%)
Expert	92	98	6
I	87	92	5
II	78	89	11
III	76	83	6
IV	71	93	22
V	69	77	8
VI	63	92	29
VII	58	89	31

Table 2–1. Seascape: Human subject classification results.

Guess	Correct class			
	Sail	Motor	Buoy	Else
Sail	26	0	0	0
Motor	4	20	2	1
Buoy	7	0	5	1
Else	3	8	3	20

(a) Experiment A (71% correct)

Guess	Correct class			
	Sail	Motor	Buoy	Else
Sail	35	0	3	0
Motor	0	21	0	2
Buoy	0	0	13	0
Else	0	8	1	24

(b) Experiment B (93% correct)

Table 2–2. Seascape: Confusion matrix for human classifiers.

Table 2–1 shows a marked increase, as expected, in classification performance in experiment B where the extra contextual information is provided to the subject. Without contextual

information many of the misclassifications are between buoys and sailboats, and also between motor boats and clutter. This is an early indication of the problems in discriminating objects within the seascape database. An important final note is that even experts classify incorrectly occasionally, and that perhaps a module that will provide an assured 100% accurate classification rate will not be possible!

## **2.7 Review**

This chapter has outlined various concepts concerning pattern recognition from both a statistical and also an overall ATR system perspective. The OSTRICH system was introduced as an example of such a system and several problems, that will be addressed in this thesis, were addressed. It has also outlined the main problem of identifying infrared objects in the seascape images and remarked on the simplicity, or not, of object recognition. The next chapter processes and analyses the seascape image database in an attempt to automatically detect and extract objects for classification.

---

## Chapter 3

# IR object segmentation, analysis and preprocessing

---

To design an enhanced object feature extraction and classification stage for a real-world ATR system required explicit knowledge of the data the system, typically, would encounter. This included the type of environment, the definition of an object, their attributes and qualities, how they were normalised, as well as, their detection, and isolation from any background sensor data. Other issues included any assumptions made in the generation of the objects, and how accurately these operations were performed.

This prior knowledge is not only helpful, but often essential in guiding a designer to realistic solutions and conclusions, and often is completely ignored in many recognition systems. The importance of this type of information is listed below.

- Provides preliminary guidance on the system design. For example, what type of sensor to incorporate for reliable object detection, or what features may prove successful in classifying the objects, and which will be bound to fail, due to a known preprocessing operation.
- Relates system performance, and output, directly to tangible, possibly physical, object characteristics, and consequently allows feedback into the design. For example, a flower classification system may fail to discriminate between two similar roses because colour has not been included as a feature. Also, it allows for the reasoning of individual object classification failures, such as when attempting to identify a rose with no petals.
- Identifies potential problems for later processing stages, for example, one highly populated class of object may dominate classifier parameter estimation.
- Alerts a designer to check the effects of failures in any of the assumptions made in the generation of the objects. For example, a particular climatic season may have been assumed.

This chapter details the prior knowledge that was available from the seascape database, and is divided into three sections; segmentation, analysis, and preprocessing. Figure 3–1 shows how this relates to the OSTRICH ATR system, outlined in the previous chapter.

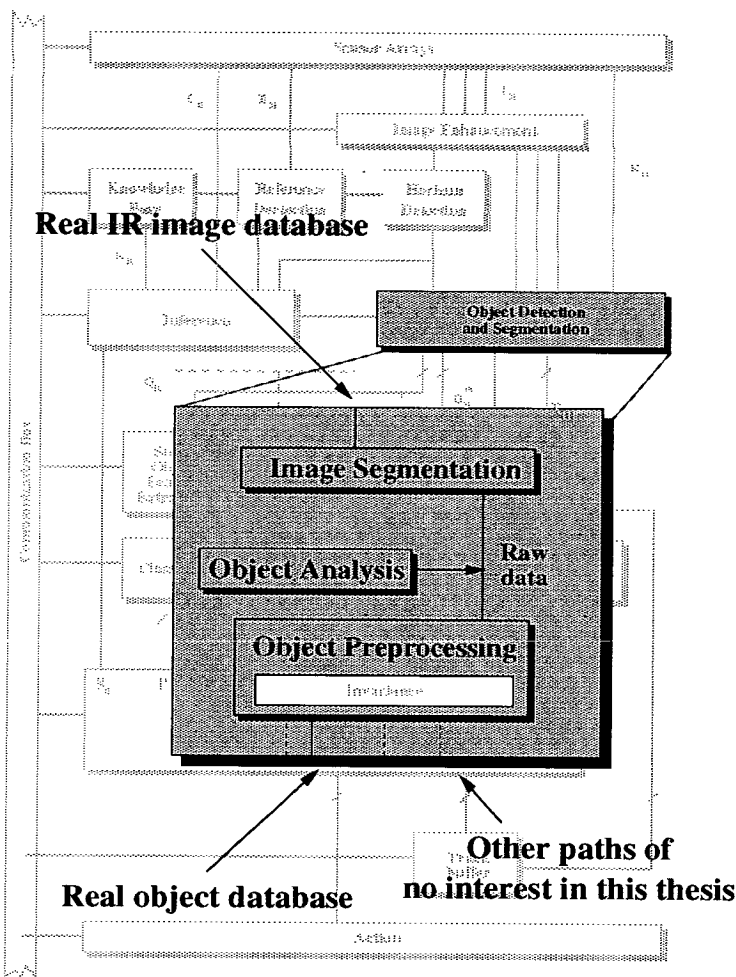


Figure 3–1: OSTRICH: Creation of an object database.

### 3.1 Object segmentation

Segmentation is the process of detecting, isolating and extracting sections of interest in a data signal for further analysis. For this to be feasible, these sections must possess some form of distinguishing localised homogeneity. For example, in an IR-based ATR system, this homogeneity exists as a strong thermal signature typically emitted by a man-made object.

In any purely object-based automated recognition scheme, the ability to accurately perform this segmentation process, such as identifying single word units from speech (*phonemic isolation*) or handwritten characters from a page of written text (*field isolation*), is fundamental to the success of any subsequent analysis, for the only indicators of object identification, given an uninformative background for all objects, are encapsulated within the segmented boundary. Shustorovich and Thrasher state, in their application of character recognition, that "*....segmentation problems account for approximately 70% of all classification errors.*" [112]

So, for the purpose of this thesis, the segmentation system was not completely automated and the segmentation parameters were adjusted manually, on a frame by frame basis, in order to produce as many accurately segmented objects as possible. Therefore, a simple gradient based operator, a *Sobel* filter, combined with various morphological processes, was sufficient. But to test the recognition system with more realistic data where the segmentation process *is* automated and non-ideal, a secondary object database of poor segmentation quality was also generated. This secondary database was created in parallel with the accurately segmented data, as within each frame it was exceedingly difficult to set the segmentation parameters such that all objects were correctly extracted.

The following sections outline the Sobel segmentation algorithm, the generation of the seascape object databases and some specific problems that were encountered with segmenting objects from this type of image database.

### 3.1.1 Sobel-based segmentation

The Sobel-based segmentation module, depicted in Figure 3–2, was designed for IR image segmentation, and was implemented in the OSTRICH system [114,115]. It consists of four basic processing units; Sobel intensity discontinuity detection, boundary detection, object determination and filtering.

Many such segmentation systems rely upon discovering the similarities and discontinuities that occur within an image,  $f(x, y)$ , where strong edges denote object boundaries. The Sobel filter is an image gradient operator and is used for edge detection [46]. It assumes regional homogeneity around transitions, with the magnitude of the local derivative operator,  $\nabla f(x, y)$ , used to detect the edges. This operator is defined as

$$\nabla f(x, y) = \left[ \frac{\partial f(x, y)}{\partial x} \quad \frac{\partial f(x, y)}{\partial y} \right]^T = [H_x \quad H_y]^T. \quad (3.1)$$

The magnitude of  $\nabla f(x, y)$ , usually is approximated by the sum of the magnitude of two directionally-dependent local derivative operators,  $H_x$  and  $H_y$ , which are calculated by passing the pair of 3x3 spatial masks, shown in Figure 3–3, across the image. These masks are known as Sobel operators, and the resulting transformed image is known as an edge map.

Generating edge maps with the seascape image database produced its own particular problems, especially when processing objects that were near, or even straddled, strong natural edges such as an horizon, or shoreline. These edges, erroneously, would be treated as part of the required object. Careful selection of the segmentation parameters though was not sufficient to minimise the effect of these edges, so a vertically orientated, low pass filter, shown in Figure 3–4, was applied to the edge map. This filter reduced the edge strength of very thin horizontal lines, such that later processes were able to remove the horizons completely.

Once the edge map was generated, it was necessary to detect the strong edges that denoted object boundaries. A two-step process was applied. First, a grey-value histogram based on the edge map was created and using a suitable threshold, the top percentage of edge pixels were set to unity, the rest to zero, generating a binarised version of the edge map. The application of this method to the seascape data was a crude but effective step, though choosing the correct threshold value proved to be difficult. To ensure that all objects in an image were detected and extracted, a reasonably high threshold was set but this subsequently extracted significant



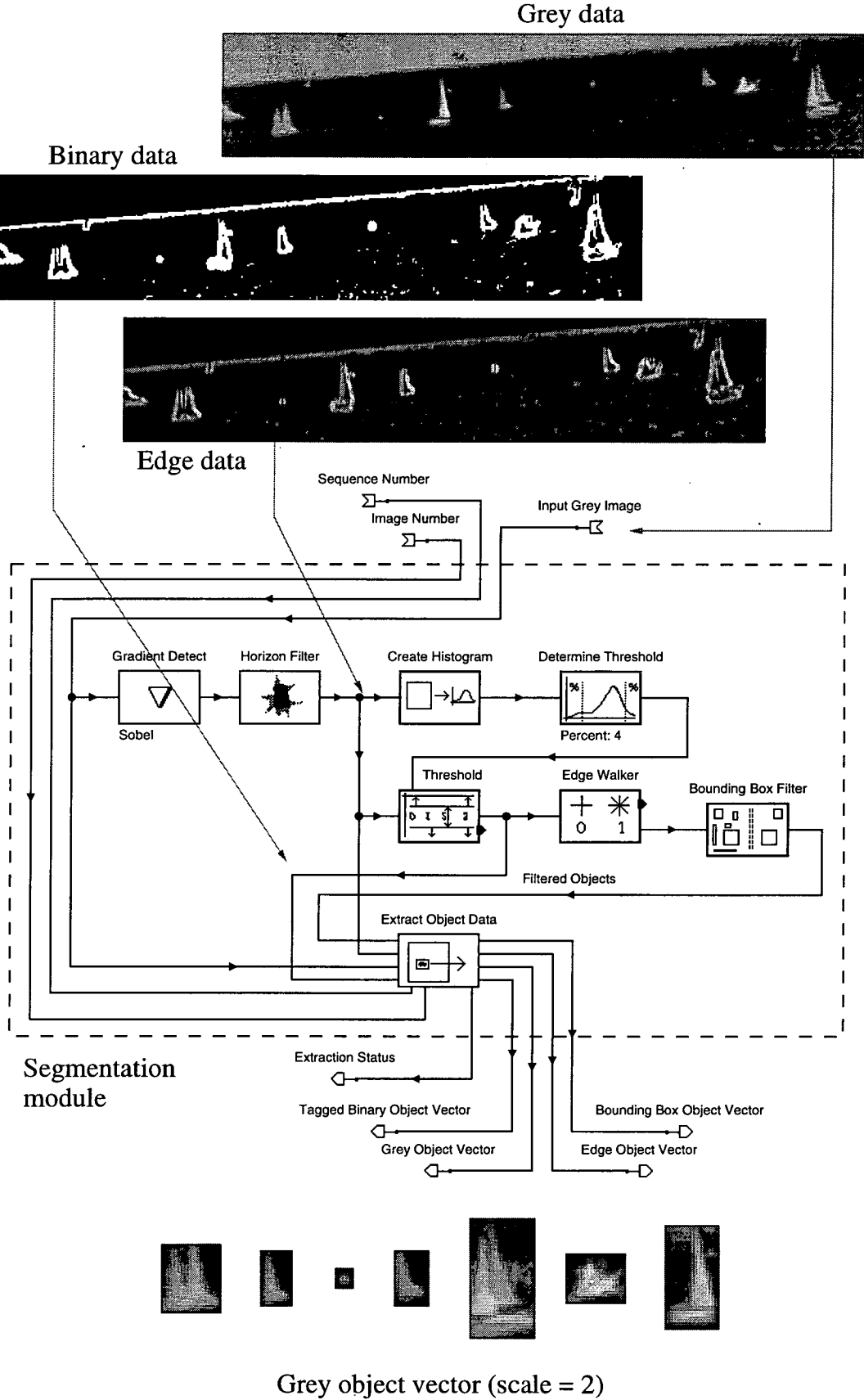


Figure 3–2: OSTRICH: Sobel segmentation module.

1	0	-1	1	2	1
2	0	-2	0	0	0
1	0	-1	-1	-2	-1

Figure 3–3: Sobel masks for generating  $H_x$  and  $H_y$ .

0	1/3	0
0	1/3	0
0	1/3	0

Figure 3–4: Horizon filter.

amounts of clutter to process and, more importantly, introduced the possibility of objects in near proximity merging. Conversely, setting the threshold too low would lead to under-segmentation or, even worse, undetected objects.

There are many, more standard, approaches for generating binarised edge maps. One approach includes the local processing of the edge data using both derivative magnitude,  $|\nabla f(x,y)|$ , and directional,  $\nabla \hat{f}(x,y)$ , information. The second approach is global and uses techniques such as the Hough transform or graph-theoretic principles [58]. The OSTRICH method was already available and there was no time to test these standard, more efficient, algorithms.

Next, the binarised edge map was operated on using an 8-way directional edge walker. This morphological process identifies an object as a connected group of unitary pixels in the binarised edge map. The edge walker makes the assumption that if the central pixel in a 3x3 grid is unitary then that pixel is part of the same object as any other unitary pixel in the surrounding 8 pixels. Again, this was crude but effective step with the seascape data.

Once detected the interior of each closed boundary binary object was filled. This generated an object mask which was subsequently tagged with an identification number. Then, each tagged object was bounded by the smallest possible rectangular box into which the object

could fit inside. This structure is termed a *bounding box*, and is characterised by two image co-ordinates. The bounding box vector, consisting of all the bounding boxes created in a particular frame, was then filtered in order to remove exceptionally large and small objects, as well as, objects with aspect ratios too thin or too wide, under the assumption that these objects were irrelevant. Finally, the tagged binary objects and the revised bounding box vector, were combined to extract, from the original frame data, a vector of rectangular, grey scale, object images, with their associated binary foreground masks.

With the seascape images the segmentation process parameters were chosen to extract particular types of raw pixel object data, namely sea craft. Typical values for the segmentation parameters, found by trial and error, are given in Table 3–1. These settings on a typical frame generated 40 objects, of which approximately 50% were segmented correctly.

Segmentation parameter	Value
Edge histogram threshold	4%
Minimum bounding box size	30
Maximum bounding box size	10,000
Minimum aspect ratio	0.125
Maximum aspect ratio	8

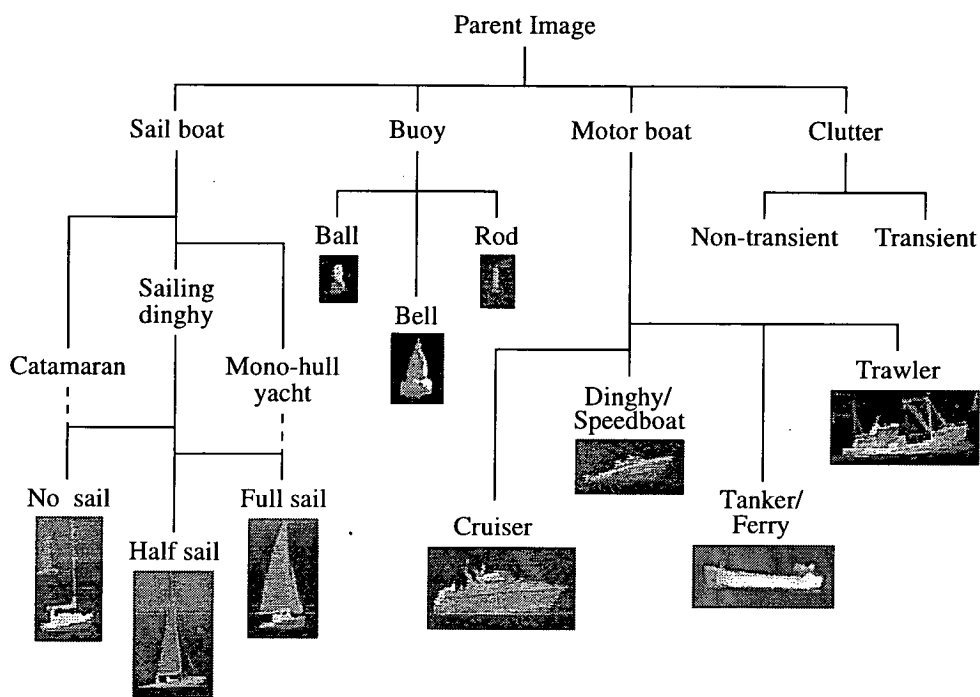
**Table 3–1:** Typical segmentation parameter values.

This data was then hand labelled and the procedure for doing this is detailed in the following section.

**3.1.2 Hand labelling of segmented objects**

To design and test classification algorithms it was necessary to possess a set of correctly labelled objects typical of those to be encountered in the final working environment. For the seascape image database the objects were labelled according to the tree structure depicted in Figure 3–5 [114].

Figure 3–5 shows objects divided into 4 main categories or classes: sailboat (class 0), motor boat(1), buoy(2) and clutter(3). In the seascape image database sailboats were found to



**Figure 3–5:** Seascape: Classification tree for the object databases.

dominate and most of the sequences contained modern sailboats, either competing in races or cruising the shoreline. These modern sailboats varied little in basic design but the ability to hoist or lower one or more sail could alter their shape significantly. Three sail states were considered adequate: no sail, full sail (all sails hoisted) and half sail (only main sail.) It was also possible for the sailboats to change considerably in shape by rotating out of the image plane. Thermally, long exposure to the sun, as in racing, or use of an inboard motor caused heating and easier detection. However, the reflective nature of white sail sheets, often a large percentage area of the object, made heat absorption very difficult and led to sails being hardly distinguishable from the background; a difficult segmentation task. Compounding this difficulty, a large thermal gradient existed between the hot hull of the boat and a very cold sea. The warmed wash around a moving boat led to further difficulties in shape definition. Lastly, it was usual for the cockpit of the boat, situated at the rear, to be the hottest section of the boat.

The motor boats existed in various shapes and sizes, ranging from simple motor dinghies to large cruisers and ferries. They were typically very hot, oblong in shape with a vertical protrusion, a cabin or perhaps a sailor, at the rear. Conceptually, each could be thought of as

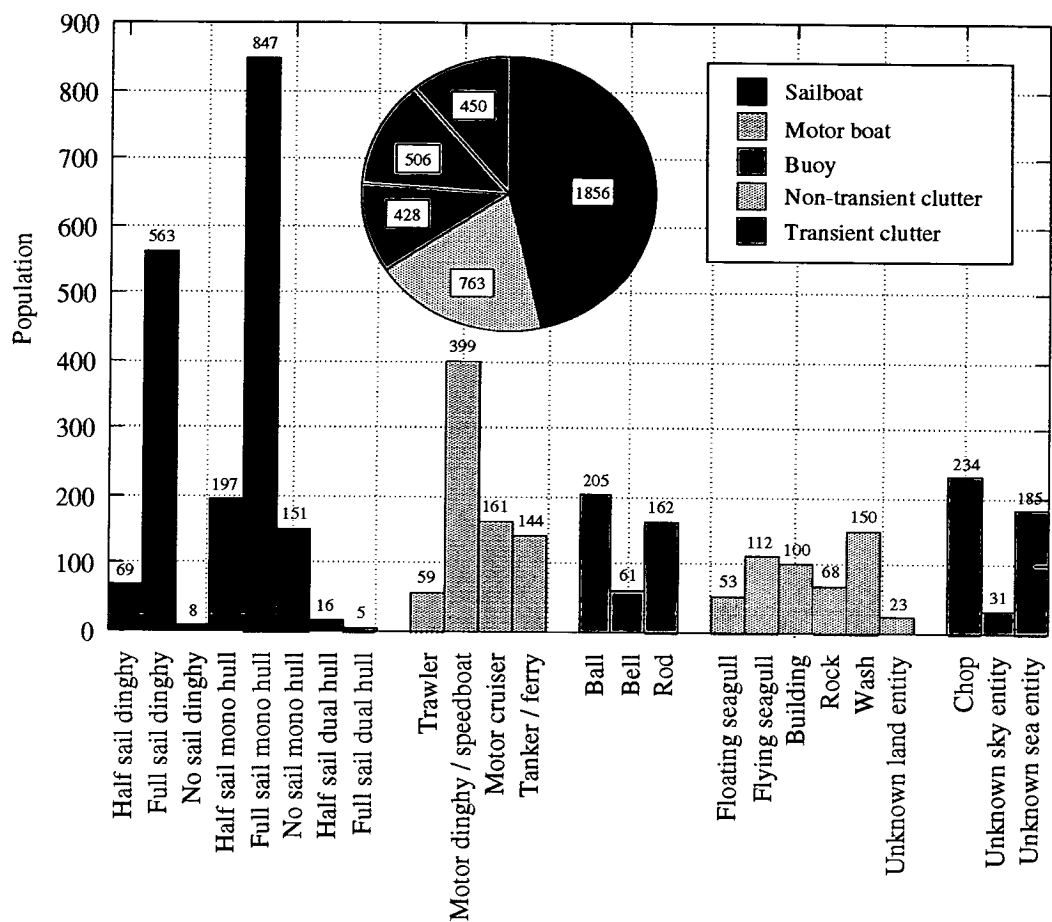
an horizontal L. Easy segmentation was hampered only by long, thin trails of wash generated by the boat. This was countered by the horizon filter in the segmentation algorithm.

The buoys fell into three distinctive categories: bell, ball and as vertical markers, or rods. They did not change shape with perspective. Mostly they were small and well defined. Unfortunately, in basic shape, at low resolution, they were easily confused with certain types of sailboat. Their distinguishing attribute was a significant thermal gradient down the buoy, the tip of the buoy being the warmest and the base, being near the water, the coolest.

The final class was the clutter class. Clutter was any object that the segmentation process extracted which was deemed, by definition of the project, uninteresting. It was often localised in space, but not necessarily in frequency, and essentially, a form of noise. Examining the clutter generated by the seascape database it was found that many were short, and wide, for example, the wash from boats or a section of coastline. One objective of a segmentation module is to minimise the occurrence of clutter whilst detecting all the objects of interest. Of course, with the sensor used in this project, it was impossible to reject all clutter because clutter could easily possess a strong thermal signature, as well as, be suitably sized. In fact, no attempt was made to improve clutter rejection in the segmentation module as this data would be required to test classifiers for their clutter rejection capabilities.

Clutter existed in two distinct subclasses; non-transient, and transient. The non-transient clutter were objects that remained stationary (for example, buildings), decayed over time (for example, wash) or could be tracked (for example, seagulls). All had some specific recognisable form. The transient clutter was assumed to be removed by a temporal classification stage, of no concern in this thesis.

Figure 3–6 shows the frequency of occurrence of each of the objects drawn from the 4003 objects extracted from the 608 seascape images.



**Figure 3–6:** Seascape: Subclass populations of all 4003 segmented objects.

3.1.3 Confirmation of the labelled data

These manual classifications were confirmed by an independent human expert and 161 (4.0%) objects were found to be manually misclassified by the original labelling. However, this included 112 objects that were comprised of multiple, connected, objects and originally classified as clutter. These were re-labelled as badly segmented objects.

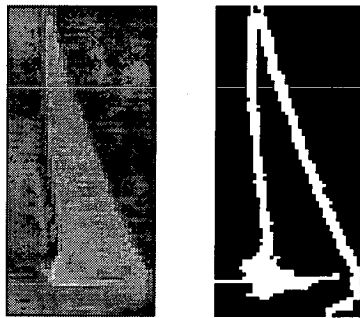
Ultimately, only 19 objects were completely discarded due to irreconcilable indecision over correct classification. There remained 3028 non-clutter objects, and they were assumed to be all correctly classified. This confirmation process highlighted two important points:

- Care must be taken in creating the original labelling scheme.
- Human classifiers are not infallible.

### 3.1.4 Seascape segmentation problems

As previously stated, two separate object databases were created from all the extracted object data. The first database contained the accurately segmented objects, for classifier experiments, whilst the second database consisted of clutter and poorly-segmented data. The poorly-segmented objects were created due to the lack of localised segmentation for each object as global segmentation parameters were not suitable for every object in a frame.

One common type of segmentation failure was non-closure. This was caused by an inappropriate choice of segmentation edge histogram threshold and was dominant in the sailboat class because of the lack of definition between the white, reflective, hull and sails, as mentioned before, and the background <sup>1</sup>. The edge map was subsequently not strong enough to denote an object boundary, and this led to the edge-walker unable to form a closed object. An example of non-closure is given in Figure 3–7.



**Figure 3–7:** Seascape: An example of the non-closure segmentation problem.

There were, of course, many other problems associated with this type of segmentation and image data [10]. These included object overlapping, similar to the problem of connected letters in character recognition, as well as horizon interference, frame interlacing, object saturation, wash from boats, and ill-defined boundaries. These problems manifested themselves as either wrongly sized bounding boxes (*external segmentation*) or distorted binary masks (*internal segmentation*), of which non-closure was an extreme case.

These problems will exist when the segmentation process finally is automated in a real-world system, even with an improved algorithm. Hence, to test the ability of the system to

---

<sup>1</sup>In FLIR it is common for parts of objects to be colder than the background [10].

identify inaccurate segmentation, all the objects in the second database were labelled, not only with their object class, but also with a measure of segmentation accuracy. Unfortunately, there was no time to develop a quantitative measure of segmentation accuracy, so a qualitative score, based on experience was implemented.

The objects were classified by both their internal (IN), and external (EX), segmentation quality with a score between 0 and 3: good segmentation (0), too large (1), too small (2) and exceptionally poor segmentation (3). The results for the sailboat, motor and buoy classes are shown in Table 3–2 with the final matrix representing the total for all three types of object. The database of well-segmented objects was determined by the number of objects with both good internal, and external, segmentation (IN0 EX0).

Sail						Motor					
EX	IN				Total	EX	IN				Total
	0	1	2	3			0	1	2	3	
0	738	169	107	486	1500	0	533	45	11	75	664
1	44	15	7	44	110	1	59	10	1	10	80
2	23	28	18	128	197	2	2	0	1	8	11
3	16	4	1	17	38	3	1	1	0	3	5
Total	821	216	133	675	1845	Total	595	56	13	96	760

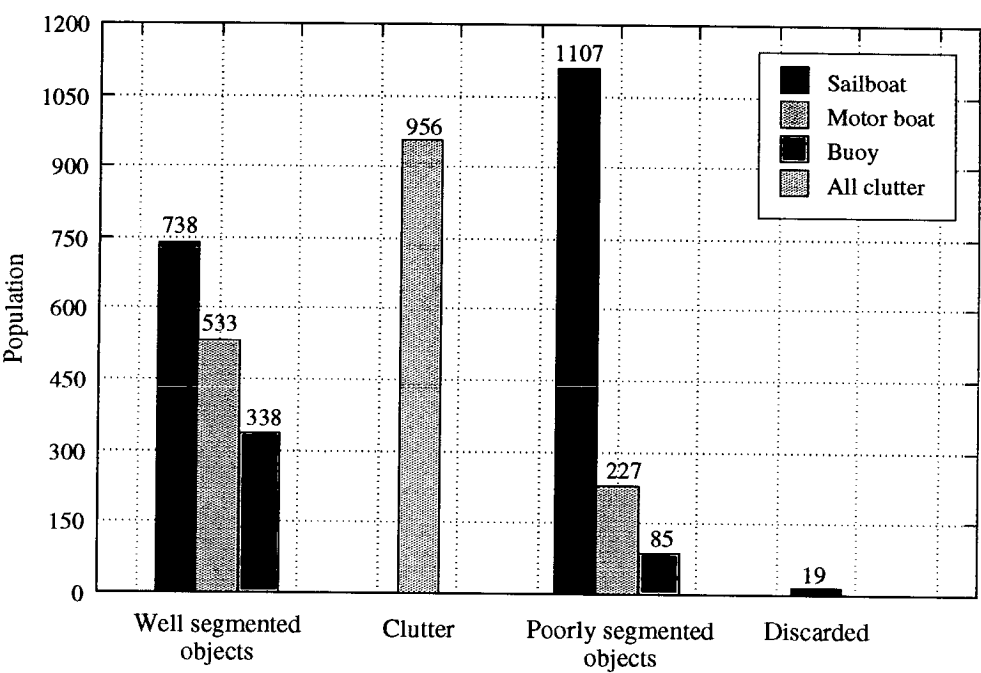
Buoy						All					
EX	IN				Total	EX	IN				Total
	0	1	2	3			0	1	2	3	
0	338	47	6	7	398	0	1609	261	124	568	2562
1	8	13	0	2	23	1	111	38	8	56	213
2	0	1	0	0	1	2	25	29	19	136	209
3	0	0	0	1	1	3	17	5	1	21	44
Total	346	61	6	10	423	Total	1762	333	152	781	3028

Table 3–2. Seascape: Segmentation quality of all non-clutter objects.

The results showed some interesting statistics specific to the database and segmentation process: non-closure of sailboats (EX0 IN3 [26% of sail class]), elongated motor boats due to wash (EX1 IN0), and loss of sail and mast with sailboats (EX2 IN0). The last-mentioned was especially interesting as a sailboat without a mast is effectively a motor boat and would be misclassified.



The results also determined the amount of well-segmented data available. If the following stages of the ATR system were not to use the internal object mask, and only require good external segmentation, there would be 2562 well-segmented non-clutter objects available. However, if the internal mask was needed this would reduce the well-segmented database to only 1609 objects, 53% of the original raw database. The latter case is summarised in Figure 3–8 where it is shown that sailboats, with their white sails, thin masts, and high probability of straddling strong natural edges such as horizons, were the hardest class of object to segment, and most prone to segmentation failure.



**Figure 3–8:** Seascape: Review of the 4003 segmented objects.

For completeness, the following section lists some of the other segmentation algorithms that were available. Many of these processes could have provided improved segmentation quality but as stated earlier using the Sobel-based approach allowed for the generation of as much rogue data, as well-segmented data, in order to fully test any new ATR classification system; under the assumption that any real ATR segmentation module will always produce some rogue objects.

### **3.1.5 Other segmentation techniques**

There are many techniques for performing object detection and segmentation. Sobel-based segmentation, which performs both these actions, works best on images with sharp intensity transitions and relatively low noise. Zero-crossing operators such as the 2-D Laplacian of a symmetric Gaussian can offer reliable edge location and tend to perform better where edges are blurred, or in noisy images. However, these have a much higher computational cost [46].

A good introduction to modern segmentation techniques can be found in a review by Pal and Pal [4]. These include simple grey-level histogram and thresholding approaches, as well as spatial filtering, boundary-based approaches, clustering, template matching, motion-based routines, fuzzy sets, Markov random fields and the use of neural network architectures [59,4].

There have also been many approaches to the specific problem of ATR segmentation utilising either single frame with range data or more advanced motion-based, multi-sensor, systems working on sequences of multi-spectral data [10,132,79]. The actual algorithms range from simple spatial filters, wavelets or texture analysis to more complicated, and often neural techniques, such as Ruck's Doppler segmentation; Tong's range segmentation algorithm utilising conditional neighbourhood filtering [129]; scanning supervised learning segmentation; and many cortical-based models.

There are also many methods purely for detecting an object. These include the hit-miss transform (HMT), wavelet transforms (for example, Haar) and hierarchical distortion-invariant filters [18].

3.2 Object analysis

There were many ways in which to characterise the seascape objects. Information could be representational, describing the attributes of a particular class, or subclass, or discriminatorial, describing the differences that exist between classes, or subclasses within a class. At the start of the chapter it was stated that in order to design a classification system information of this type could be very beneficial. This section describes the empirical measures that were employed to characterise the well-segmented seascape objects. These measures hoped to highlight the similarities and disparities that exist between objects in the database, noting useful discriminatorial features, and identifying sources of misleading or over-optimistic information. Table 3–3 lists the features found useful for describing the well-segmented seascape objects in this way. The characteristics were divided into five levels, according to the type of object data from which the features were derived. These levels included bounding box, object outline, binary mask, grey-level pixel data and abstract level.

Bounding Box	Outline	Binary	Grey	Abstract
Width	Bending energy	% foreground	Centre of mass	Rotation
Height	Compactness		Symmetry	Temporal
Aspect ratio	Elongation		Pixel value	
	Corners		Texture	
			Histogram	

Table 3–3. Seascape: Object characteristics divided into five levels.

3.2.1 Bounding box analysis

This basic level of analysis provided information regarding an object’s position, population and relative size. The position of an object, either relative or absolute, itself can not provide directly any clues to object identification but can be used in later ATR interpretation stages, combined with knowledge base data and tracking information, to improve greatly classification reliability. The number of objects detected in a frame was also irrelevant for individual object

classification. Though, as with object location, could possibly provide information to the ATR interpretation stage leading to a different course of action being taken.

The relative size of all the well-segmented object bounding boxes in the seascape database is given in Figure 3–9. The height versus width plot<sup>2</sup>, shows distinct divisions existing between the classes with respect to height, width and aspect ratio. Sailboats tend to be tall and thin, and range broadly in pixel size, as shown in the object size frequency plot, also in Figure 3–9. Motor boats are similarly distributed but are, generally, much wider than they are high. Buoys are smaller in size with aspect ratio's close to unity. Finally, the clutter, like the motor boats, tends to be short and wide, but unlike the motor boats are mainly small in total pixel size.

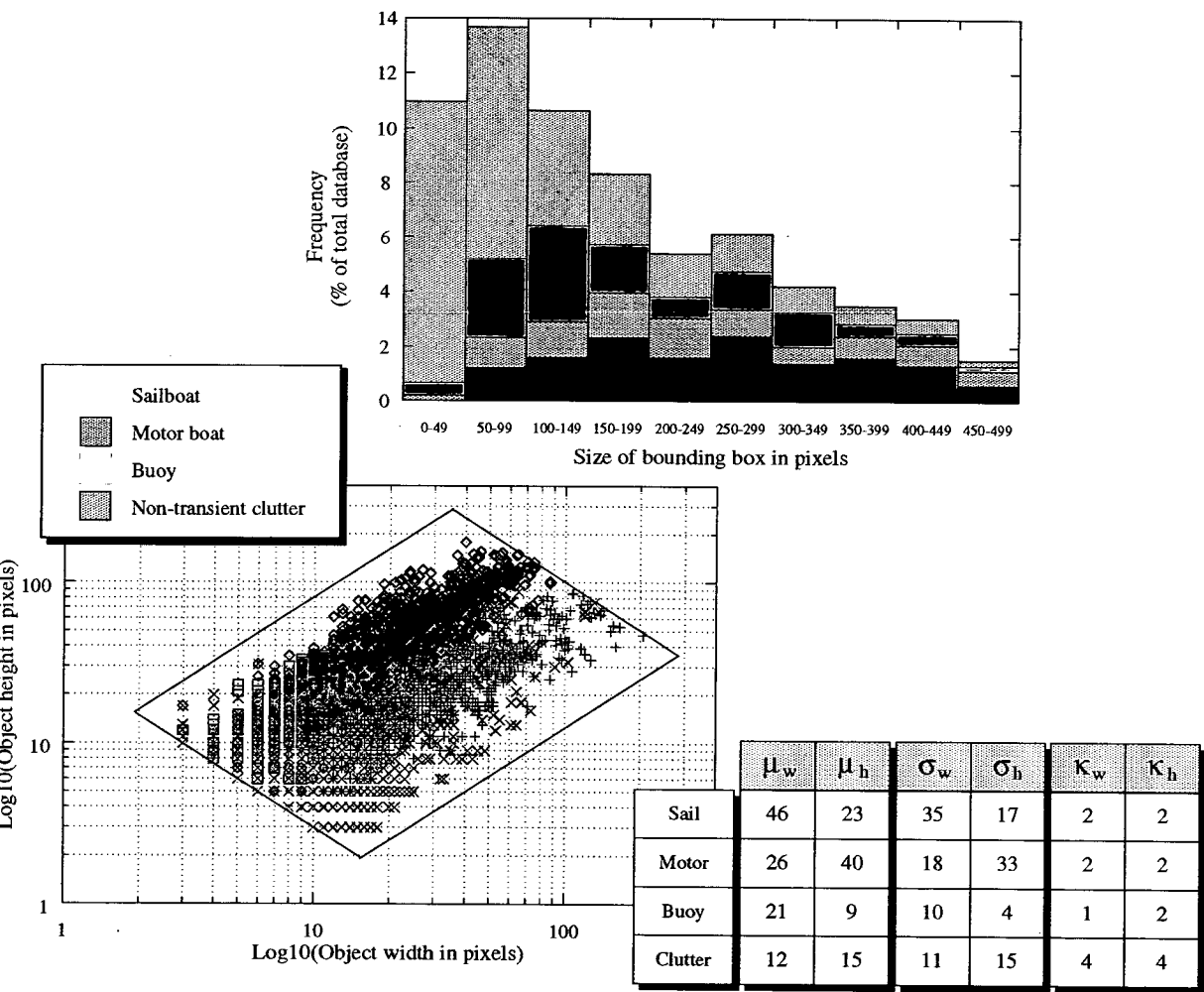


Figure 3–9: Seascape: Distribution of well-segmented, object sizes.

<sup>2</sup>The size of the parallelogram is controlled by object size and aspect ratio filter segmentation parameters.

The table in Figure 3–9 provides mean ( $\mu$ ), standard deviation ( $\sigma$ ), and skew ( $\kappa$ ) pixel values with respect to width and height, for each of the four classes.

Unfortunately, the size of an object's bounding box is a function of an object's range, as well as, physical size. As range data was unavailable with the seascape data, size information had to be ignored, or normalised. For example, an unlabelled test object would be classified as either clutter, or buoy, simply because it was small although it may have been simply a sailboat in the distance. In a practical ATR system identification of a potential target must occur as soon as possible, when the object *is* in the distance, and not when the object close enough to constitute a threat. So, object size, without range data, is useless but aspect ratio, which is not a function of range, was found to be a useful discriminative feature.

### 3.2.2 Outline analysis

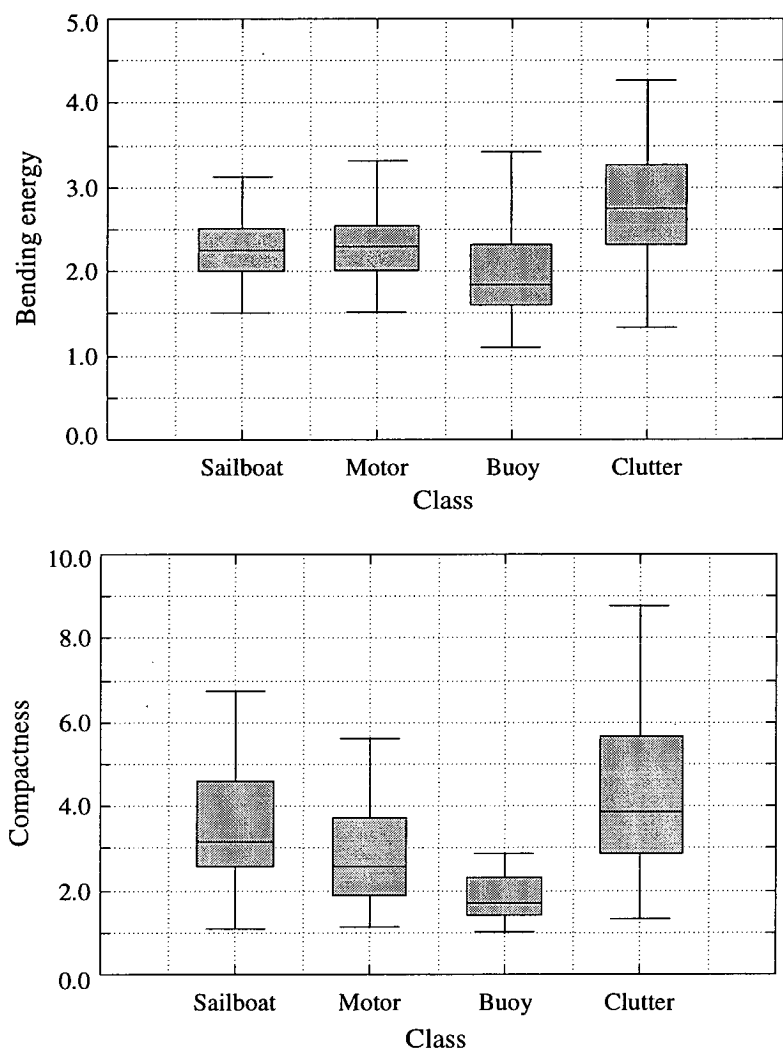
The segmentation process, discussed earlier, generated two binary images: an object boundary, and an object mask. The boundary defines the shape of the object. A visual inspection of the database suggested that most object outlines consisted of mainly low-frequency components. Four descriptors, often used to describe the outline of an object, were applied to the seascape data. These included bending energy, compactness, elongation, as well as the number of corners.

Bending energy measures the twistedness of an outline. If the curvature at a point  $t$  along an object's boundary of length  $T$  is defined as  $|\kappa(t)|^2 = (d^2x/dt^2)^2 + (d^2y/dt^2)^2$  then the total bending energy of the object is  $\int_0^T |\kappa(t)|^2 / T dt$ . A similar attribute,  $\gamma = T^2 / 4\pi(\text{area})$ , measures object roundedness, or compactness. Hence, for a circle  $\gamma = 1$  [58].

The results, given in Figure 3–10, show that buoys tend to be simplistic in shape and approximately circular, whilst the motor boats and sailboats are more complex. Clutter, as was expected, displays a broad range of convoluted boundaries.



The number of corners in an object was measured by thresholding the curvature,  $|\kappa(t)|$ , at some suitably large value. This is demonstrated in the top plot of Figure 3–11 where the sailboat is seen to have three distinctive corners; A, B, and C.



**Figure 3–10:** Seascape: Box-plots for bending energy and compactness in each class.

The application of this technique though was found to be unreliable, but generally supported the view that buoys are typically round to triangular, sailboats definitely triangular and motor boats rectangular.

The left-lower plot in Figure 3–11 shows the normalised radial distance from the centre of mass to various points along the boundary. The ratio  $\rho_{max}/\rho_{min}$  was useful as a measure of object elongation, similar to bounding box aspect ratio.

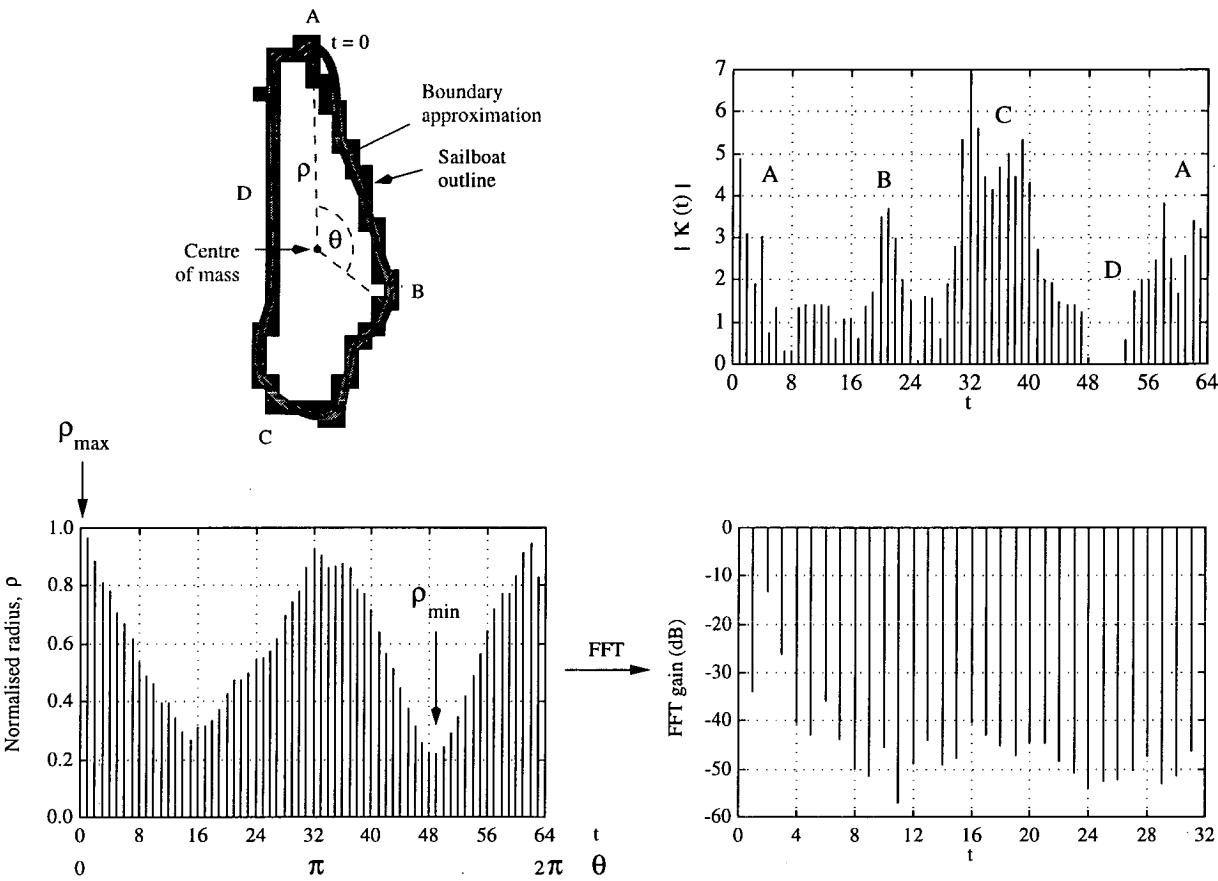


Figure 3–11: Seascape: Curvature and  $(\rho, \theta)$  plot for a sailboat outline.

3.2.3 Binary mask analysis

The binary mask covers one object within the associated set of bounding box co-ordinates and was used to determine the percentage foreground of an object within the limiting rectangle. Knowledge of the amount of area covered by an object within the confines of its bounding box was essential in assessing the effect, or risk, of interference from overlapping objects in the background.

Table 3–4 provides the mean percentage foreground estimates for each class of well-segmented object. The large percentage background with the sailboat class was attributed to the triangular nature of boats with sails, and more particularly to boats with only a mast, for example as shown in Figure 3–17.

Class	Mean (%)	Standard deviation (%)
Sail	55.7	12.3
Motor	69.4	9.3
Buoy	77.1	8.5
Clutter	52.8	15.8

**Table 3–4.** Seascape: Mean percentage of bounding box area filled by object.

3.2.4 Grey-level analysis

The grey-level value of each object pixel is dependent on the temperature of the heat source. Unfortunately, this value is also a function of the overall number of heat sources in the scene. This is due to the variable thermal window discussed in Chapter 2. However, this dependency upon the number of objects in each frame was found to be not severe and the small linear shifts introduced into the grey-level histograms could be compensated for by suitable normalisation. Figure 3–12 shows foreground, grey-level, histograms for three different well-segmented objects, given as a percentage of the total number of pixels in each image.

Attributes are often derived from grey-level histograms to distinguish between foreground and background but, as can be seen, the distribution of data is similar for each of the three non-clutter classes; broad, multi-modal and often saturated at the maximum grey-level value of 255. To discriminate between the classes, an analysis of the spatial relationships of the grey-levels was more appropriate. These analyses ranged from localised features such as texture, to more global attributes such as symmetry and centre of mass. Texture has been shown to be an excellent feature for determining identity in research areas such as remote sensing [53]. In these applications the shape of the object is often irrelevant; for example the shape of a field is usually no indicator of the vegetation. However, in most ATR problems, for distinguishing between



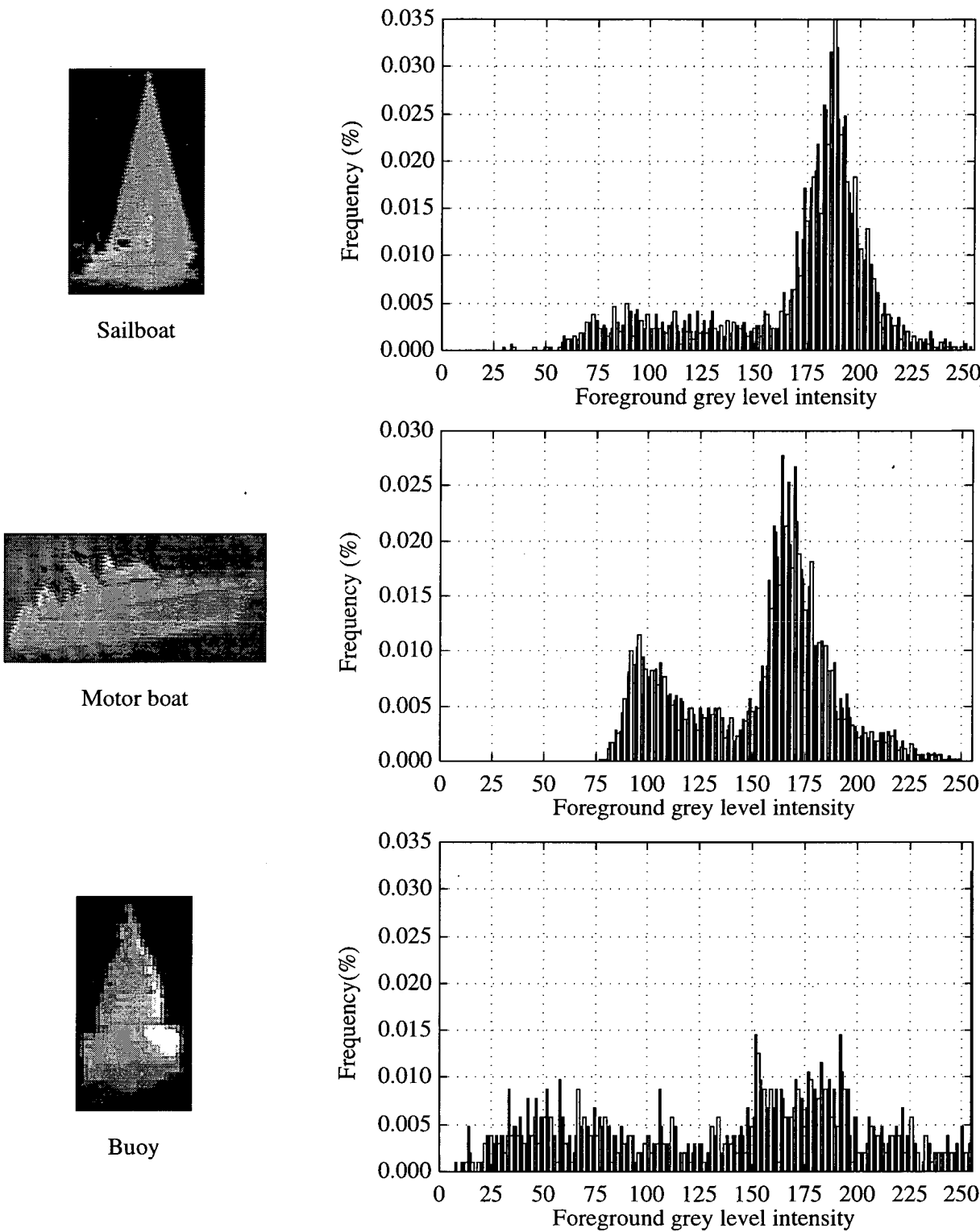
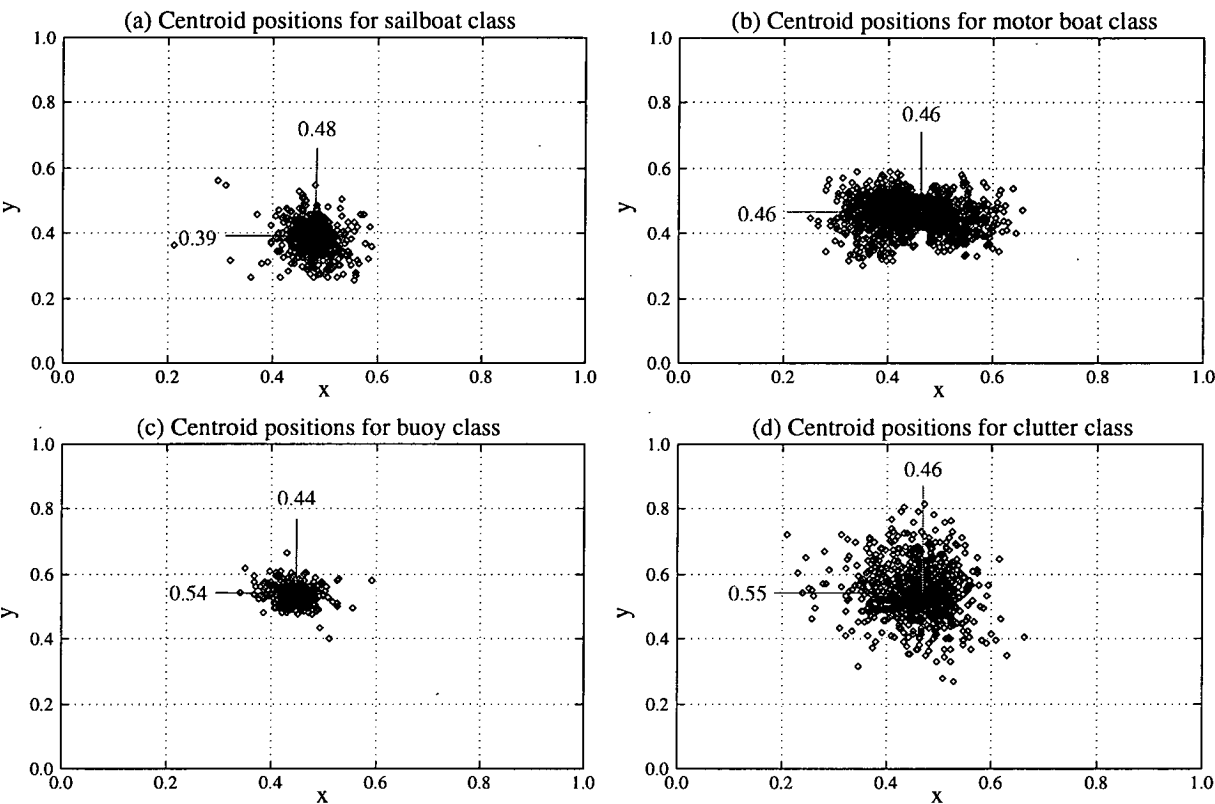


Figure 3–12: Seascape: Foreground grey-level histogram for a sailboat, motor boat and buoy.

foreground object classes, general shape and energy distribution is crucial. Consequently, the distribution of pixel values across the object was an important issue. For example, as previously stated, the main heat source on a motor boat is the engine at the rear of the craft.

The lowest order image moment is the centre of mass  $(\bar{x}, \bar{y})$ , or centroid, measure [122]. Figure 3–13 plots the normalised centroid positions,  $(\bar{x}/width, \bar{y}/height)$ , for each class. The

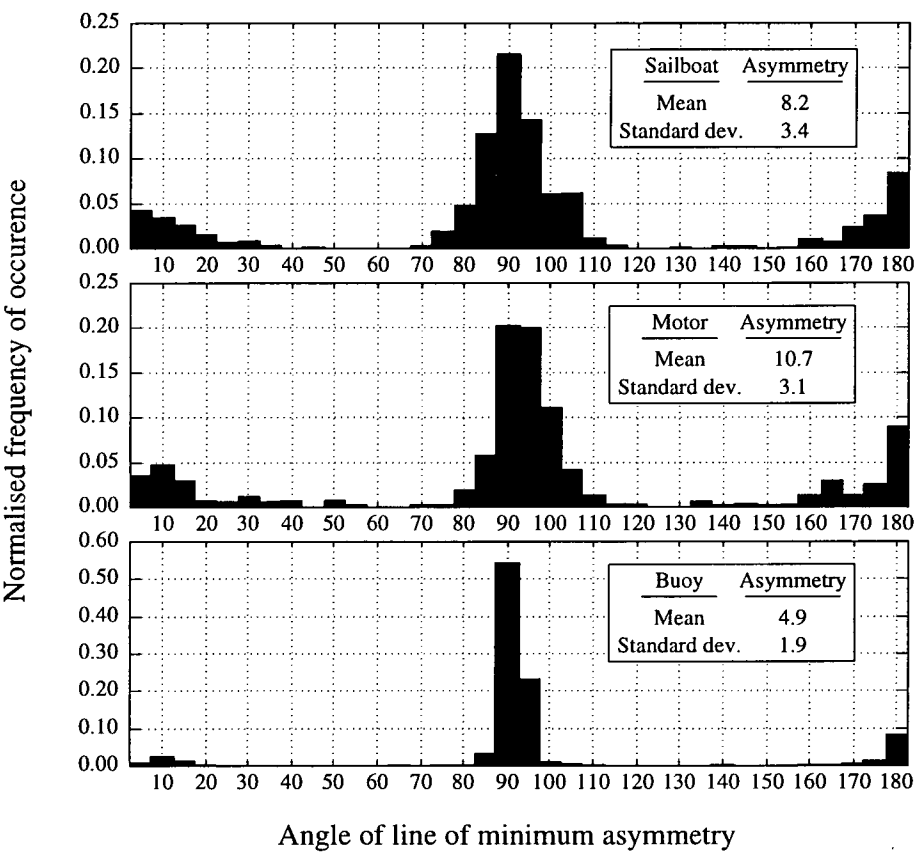


**Figure 3–13:** Seascape: Class normalised centroid distributions.

sailboats and buoys show little variance in centroid position. For the sailboat data the mean centroid value was found in the lower regions of the object, due to the hot hull and the tall, cool, white sails. For the buoy the opposite was true, though less extreme, with more mass in the upper regions due to the thermal gradient down the buoy, as mentioned earlier. However, the motor boats and clutter exhibited highly variant centroids. This variance was expected for the clutter class, but not for the motor boats. It was realised that motor boat centroid distribution was at least bimodal, in that the centre of mass would depend on the direction of travel. Travelling left-to-right meant the large mass of the hot engine would move the centroid towards the left, and vice-versa. Excluding clutter, these observations indicated that centroid

position was a good indicator of object classification; centroids shifted down for sailboats, left or right for motor boats and a slight upward shift for buoys.

Another grey-level analysis performed measured object symmetry. Figure 3–14 shows the distribution of the angles of minimum asymmetry (maximum symmetry) for each non-clutter class across the seascape database. A line of minimum asymmetry passes through the object centroid and is orientated such that an asymmetry measure is minimised. The asymmetry measure used with the seascape data calculated the absolute difference between a point,  $p$ , and its mirror image,  $p'$ , through the line of asymmetry, summed over the image. For an



**Figure 3–14:** Seascape: Angles of minimum asymmetry (maximum symmetry.)

8-bit, grey-level, image an asymmetry value of 128 represents complete asymmetry and 0 exact symmetry. The mean and standard deviation of the asymmetry values in the upright position are given in Figure 3–14. In the upright position (90 degrees), the objects were, unsurprisingly, most symmetric. The mean asymmetric values indicated that, at 90 degrees, the motor boats

were the least symmetric. This, again, was due to the heat of the engine in the left, or right, extremities of the image. The buoys exhibited the highest amount of symmetry.

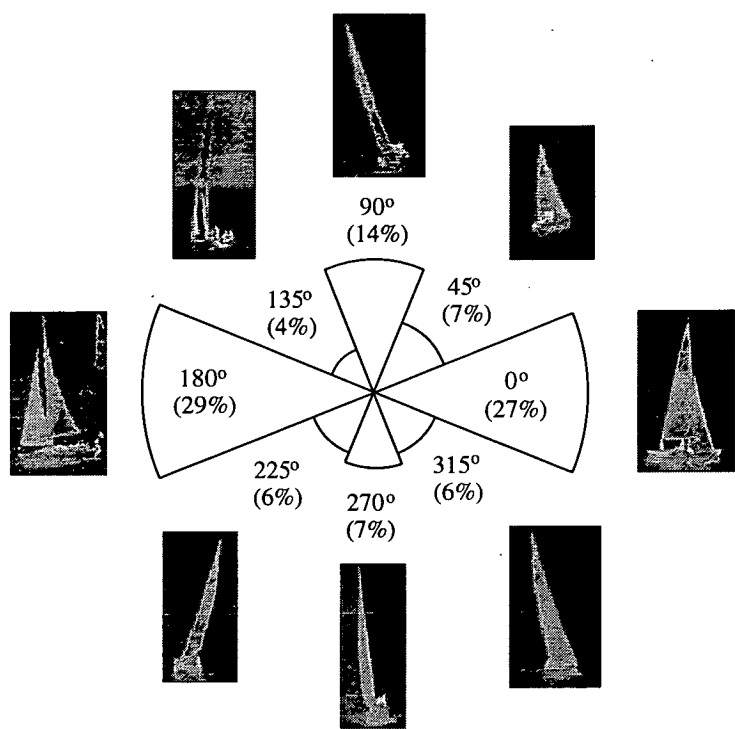
Returning to the actual distribution of minimum asymmetry angles in Figure 3–14 there exist greater deviations about 90 degrees with both the motor boat, and sailboats. This tilting, or rotating, was found in both objects when travelling at speed. The power of the engine pushed the nose of a motor boat up, and the combination of the wind and sharp turning manoeuvres pushed the sailboats over. An example of this is provided in the 90 degree image in Figure 3–15.

### **3.2.5 Abstract level analysis**

The final level of analysis considered the object in the original image and, in particular, its three dimensional properties. Knowledge of out-of-plane object rotations was important as two dimensional profiles often altered with this property. This was irrelevant for the buoy class as their appearance did not alter with out-of-plane rotation but boats, however, changed considerably in shape, especially aspect ratio.

Each of the objects in the seascape database was labelled with one of eight out-of-plane orientations: 0, 45, 90, 135, 180, 225, 270 and 315 degrees. This was adequate for determining the distributions of directions of motion. Figure 3–15 shows how the sailboat class is divided among the different orientations. Similar distributions were found with the motor boats, but with many more existing in the 0 and 180 degree bins (79% in total). It appeared that many object images were taken from a side-on perspective. Also, it was noted that 45, 135, 225, and 315 orientations were often very similar in overall shape to objects oriented at 0, 90, 180 or 270. Finally, objects at 90 and 270 (coming towards or away) were hard to discriminate, and both 0 and 180 orientated objects, assuming both fore and aft sails were raised, were similar by symmetry. Hence, only 2, possibly 3, orientational subclasses, in each class, needed to be considered for the boats.

A final type of analysis that could have been performed was temporal analysis, examining how objects altered over frames. This information was not available with the seascape data as contiguous frames of video were not captured. This was due to the lack of time and processing resources, and to ensure that the seascape database did not contain objects that were practically identical because they were, for example, images of the same sailboat only captured fractions of



**Figure 3–15:** Seascape: Rose diagram showing directional populations of sailboat class.

seconds apart. However, the transient clutter was assumed to be filtered out by such a temporal analysis in later stages of the ATR system.

**3.2.6 Analysis conclusions**

Table 3–5 summarises the analysis of the seascape database discussed in the previous sections. Overall, each class has excellent discriminative properties but there were similarities that could cause confusion. Furthermore, the intra-class variabilities that exist in each class, for differing reasons, could increase the complexity of the classification model.

Class	Results
Sailboat	<ul style="list-style-type: none"><li>• Most common non-clutter class</li><li>• Subclasses much alike in terms of design, but sail state considerably effects shape</li><li>• Further subclasses due to out-of plane rotation</li><li>• Triangular, tall and thin</li><li>• Mass resides in lower section of image</li><li>• Large percentage of image is background</li><li>• Difficult to segment</li><li>• Tilts 10-15 degrees when travelling at speed, or turning</li></ul>
Motor	<ul style="list-style-type: none"><li>• Mass centred towards left, or right, extremes, due to hot engine</li><li>• Rectangular, short and wide</li><li>• Subclasses due to design, and direction of travel</li><li>• Least symmetric of all non-clutter classes</li><li>• Smaller, faster motor boats tilt up when travelling at speed</li></ul>
Buoy	<ul style="list-style-type: none"><li>• Least common non-clutter class</li><li>• Round, though less so with rod buoy subclass</li><li>• Three distinct classes due to design</li><li>• High two, and three, dimensional symmetry</li><li>• Most likely to be confused with a sailboat</li></ul>
Clutter	<ul style="list-style-type: none"><li>• Rectangular, short and wide but small in total pixel size</li><li>• Two subclasses, but one filtered out in later ATR stages</li><li>• No single distinctive shape, or centre of mass</li><li>• An often convoluted boundary</li><li>• Most likely to cause confusion with the motor class</li></ul>

**Table 3–5:** Seascape: Analysis conclusions.

### 3.3 Object preprocessing

The detailed analysis of the raw object data, described in this chapter, highlighted many of the intra- and inter-class variations that exist in the seascape database. The inter-class variations are highly desirable for classification purposes, but the intra-class are highly undesirable. There were other properties that varied from object to object, such as scale, background influence and grey-level shifts, the effects of which all required suitable preprocessing, and normalisation, for reasons described earlier in the chapter. This section describes the actual preprocessing method.

The grey-level image data for each object was, initially, low-pass filtered to reduce noise and some of the high frequency artifacts peculiar to individual objects. The foreground of each object was then histogram equalised. This countered some of the effects of the variable thermal window. The next step was to scale each object, whilst preserving the aspect ratio, in order to counter, for example, changes in camera zoom. By rescaling the objects a rudimentary form of size invariance was achieved and allowed for easier handling of the object data in the feature extraction and classification stages.

Rescaling was performed by reconstructing the original image and then re-sampling at the new frequency. Theoretically, this could be performed exactly, if the image was band limited and a sinc-based kernel interpolator used [58,77]. However, in practice, the problem was to find a suitable kernel with respect to reconstruction error and computational overhead.

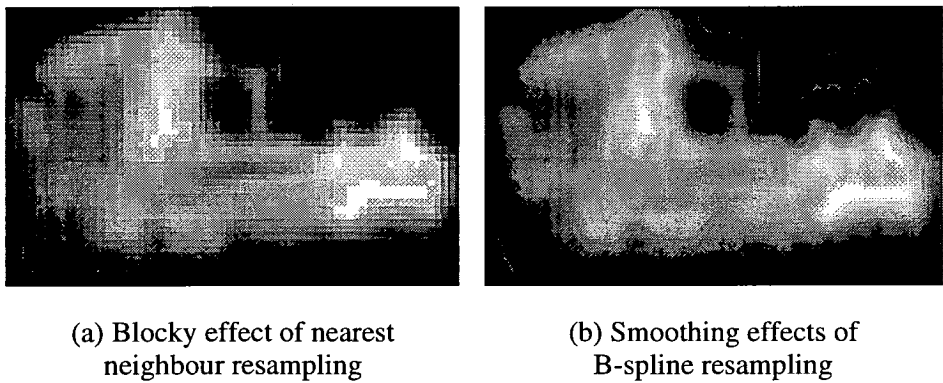
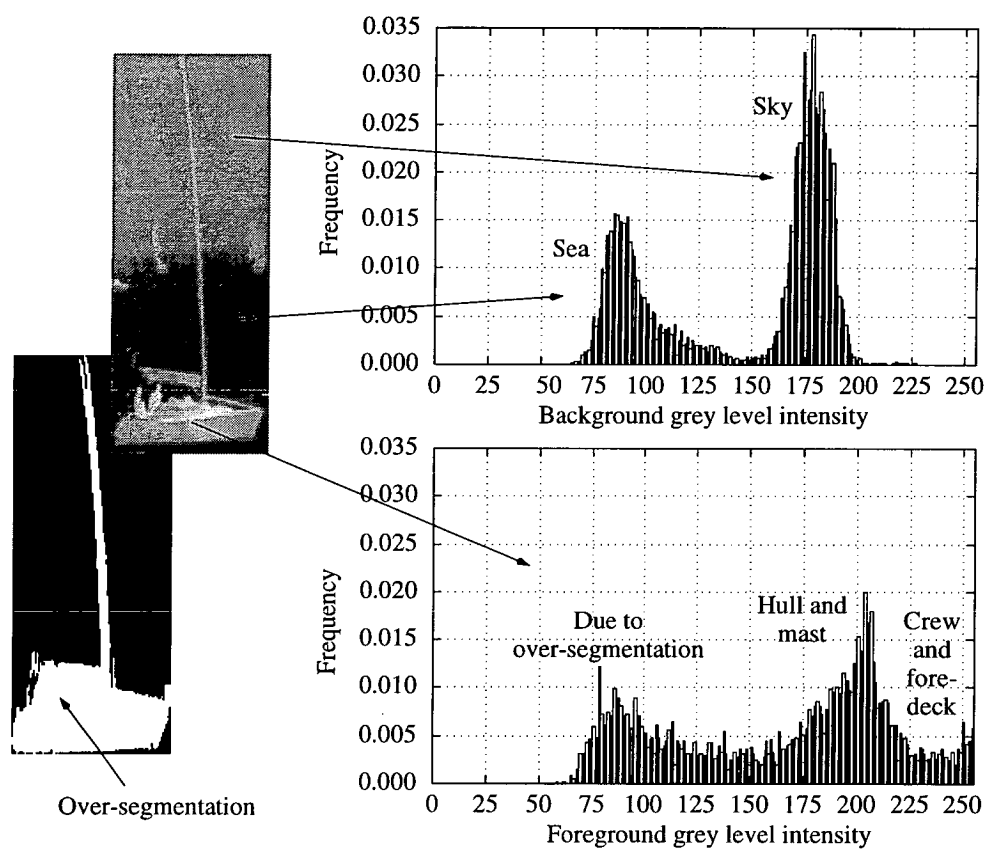


Figure 3–16: Re-sampling examples

Parker, Kenyon and Troxel provide an excellent comparison of several common interpolating methods for image re-sampling [77] including nearest neighbour, linear, cubic B-splines and high resolution cubic splines. Examples are shown, for a seascape object, in Figure 3–16. They suggest the use of the  $a = -0.5$  high resolution cubic spline as most appropriate when



**Figure 3–17:** Seascape: Grey-level histograms for a typical sailboat.

further mathematical processing of the objects is to be performed. This process was applied to both the grey and binary object images such that the objects were rescaled to either 16x16 or 32x32 pixel images, with quantisation effects reduced due to histogram equalisation. The pixel elements then were labelled from 0 to either 15, or 31, in each direction. However, to aid processing in later stages, a secondary set of labels, uniformly ranging from -1.0 to 1.0, were introduced spanning each axis of the image.

As described in the binary analysis section a large proportion, 40%, of the object image is comprised of background. This was further increased during rescaling because the bounding box was enlarged to generate a square in order to maintain aspect ratio. The probability of background interference, in a cluttered environment, was too great and for this reason the



background pixel values in the grey-level object image were set to zero. Thus, there was a heavy dependence on segmentation accuracy. Figure 3–17 demonstrates the possible influence of retaining the background pixel values. Note the two dominant, low variance, peaks caused by the sea, with low mean, and especially the sky, with high mean. An identical object, with no sky in the background, would have a completely different profile and subsequently generate possibly very different features.

A review of this section of the OSTRICH system is provided in Figure 3–18<sup>3</sup>.

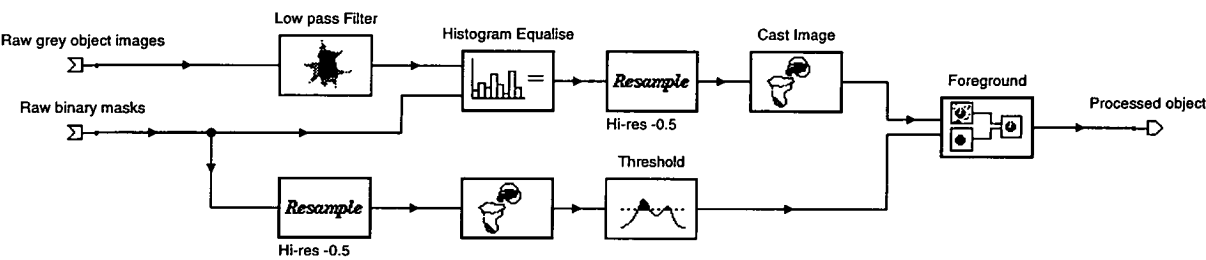


Figure 3–18: OSTRICH: The preprocessing system.

3.4 Review

This chapter has reviewed the process of generating two object databases derived from the seascape image database described in Chapter 2. The first database contained three classes of accurately segmented objects, whilst the other consisted of all the other products of this extraction process.

The chapter has examined how each of the databases were created, how each of the objects were labelled, processed, assessed for quality, and how the well-segmented data was analysed and characterised. Features were described that both represented classes of objects, features that discriminated between classes of objects, and features, such as range data, that would have significantly aided analysis, had they been available. Furthermore, all assumptions that were made in creating the data were listed. Some of the key assumptions are listed on the next page:

<sup>3</sup>The "Cast Image" operator is required by the OSTRICH system and is of no concern here.

- Object boundaries will always be well defined in the seascape environment.
- Aspect ratio filtering will not remove any objects of interest.
- No object could possibly exist in more than one class. For example, a dinghy with an outboard motor, and a sail.
- Transient clutter can be removed by later temporal processing.
- After confirmation of the data labelling, all objects are correctly classified in the database.
- The segmentation module will always generate rogue data, whether clutter or poorly segmented.
- Out-of-plane rotations of objects can be treated as a small number of subclasses.
- A constant aspect angle is used.
- The analytical tools used were appropriate.
- Object bounding box size was irrelevant without range data, and was required to normalised.
- No false, artificial, or misrepresentative information has been introduced through any of the preprocessing stages.
- And finally, object databases are representative of the real-world data to be encountered in the final operating system. This is actually a very weak assumption due to the lack of variation in the seascape image database, as described in Chapter 2.

The next chapter uses the analysis of the well-segmented data to generate sets of features, some of which were introduced in this chapter, for classification. Later chapters will make use of the other data, and information, created in this chapter.

---

## Chapter 4

# Feature extraction and classification

---

The previous chapter has described how the well-segmented objects were extracted, and normalised, from the database of real, infrared, seascape images. These well-segmented objects were now to be used as a basis to test new classification algorithms with a confidence that the objects had known characteristics. In the previous chapter it was shown that some of these characteristics, or *features*, made it possible to separate the objects from the background. Unfortunately, this rudimentary form of classification was unable to perform the finer differentiation required to determine object class. Thus, different, and often more complicated, features were required to perform the object classification.

Feature extraction is the process of mapping originally high dimensional image patterns, generated by a segmentation process, into a much lower, and manageable, dimensional subspace. This transformation is intended to remove any redundancy or correlations in the data and to reduce the number classifier inputs without significantly reducing the class separability that exists in the original space. A classifier with fewer inputs requires less model parameters to be estimated, possibly improves generalisation due to a higher parameter to size of database ratio, reduces weight storage and is faster to train. Now ideally, both the feature extraction and classification stages should be optimised together but this is often restricted by practical constraints and the two stages often have to be treated separately [13](page 305.)

The problem with both the seascape, and NIST, data was how to extract reasonably-sized, independent, sets of features for classification. This chapter discusses the standard feature extraction and classification techniques that were tested, and the statistical feature selection techniques that were applied to predict which of the extracted features would provide the easiest discrimination of classes. Furthermore, this chapter reports on the complications that were encountered due to treating the feature extraction and classification stages as separate



## 4.1 Feature extraction

The feature extraction stage, highlighted in Figure 4–1, was provided with object data from the segmentation process. Consequently, features could be derived only from within the boundary of the object. This excluded data such as geographical or environmental information, allowing for an unbiased classification based on object shape alone, at one particular instant. This would provide independent data to the later interpretation stage. The challenge was to determine a suitable set of shape-based features that would provide good generalisation.

This section explains why feature extraction was necessary, outlining the various feature extraction techniques that have been used previously for character recognition and ATR. The techniques examined were from two different sources: statistical, describing features that were derived from each object through analysis, as in Chapter 3; and linear spatial mappings whereby features were generated through linear transformations of the image data. Due to time limitations of the project, only these type of features were considered.

### 4.1.1 Determinedness

In Chapter 2 the current BASE classifier was described as having approximately 130,000 model parameters, which were estimated using a finite training database of 2000 samples. Model estimates were thus under-determined, even with the inherent self-correlation of the images. A 200 parameter model which would be faster to train, provide improved generalisation, and require significantly less storage was a far more attractive proposition. This implied a classifier with about 10 inputs.

This problem of under-determinedness is also known as *Bellman's curse of dimensionality* and is covered in many texts [13]. In short, as the number of inputs increases, the number of data samples required to define the class boundaries increases exponentially. Thus, classifiers with fewer inputs are more desirable. Cheng and Titterington comment on the generalisation ability of Le Cun's Zip-code image recognition system with a 16x16 pixel input array [22]. This had 9760 independent parameters and was trained with 7291 data samples. A significant increase in the generalisation ability was achieved when the number of parameters was decreased by

a factor of four [22]. Feature extraction is a standard method of significantly reducing the number of inputs, and subsequently parameters.

Having now discussed why feature extraction was required, the two types of feature, derived from statistical measures and linear sub-spatial mapping, are described further.

### 4.1.2 Statistical features

Table 4–1 lists 32 statistical features that have been used previously to classify successfully various object databases. Features 1 through 12 were introduced in Chapter 3. The other features are statistical measures used to describe the grey level pixel distribution of the object, also described in Chapter 3. These pixel-distribution features are reliant on the size of the pre-normalised object being large enough to produce valid distributional estimates.

### 4.1.3 Linear spatially-mapped features

Features derived from linear spatial mappings are simple to generate. Each  $M$ -dimensional feature vector,  $\mathbf{d}$ , is related to the  $N^2 \times 1$ -dimensional vector representation,  $\mathbf{f}$ , of an  $N$  by  $N$  pixel image  $f(x, y)$ , by the equation

$$\mathbf{d} = \mathbf{A} \mathbf{f} \quad (4.1)$$

where  $\mathbf{A}$  is an  $M \times N^2$  matrix and, usually,  $M \ll N^2$ . Alternatively, this can be given as

$$d_m = \sum_x \sum_y^N f(x, y) g_m(x, y), \quad (4.2)$$

where  $g_m(x, y) \forall m = 1, 2, \dots, M$  are the feature extracting kernels. Kernel selection is directly related to the quality of the features generated for classification.

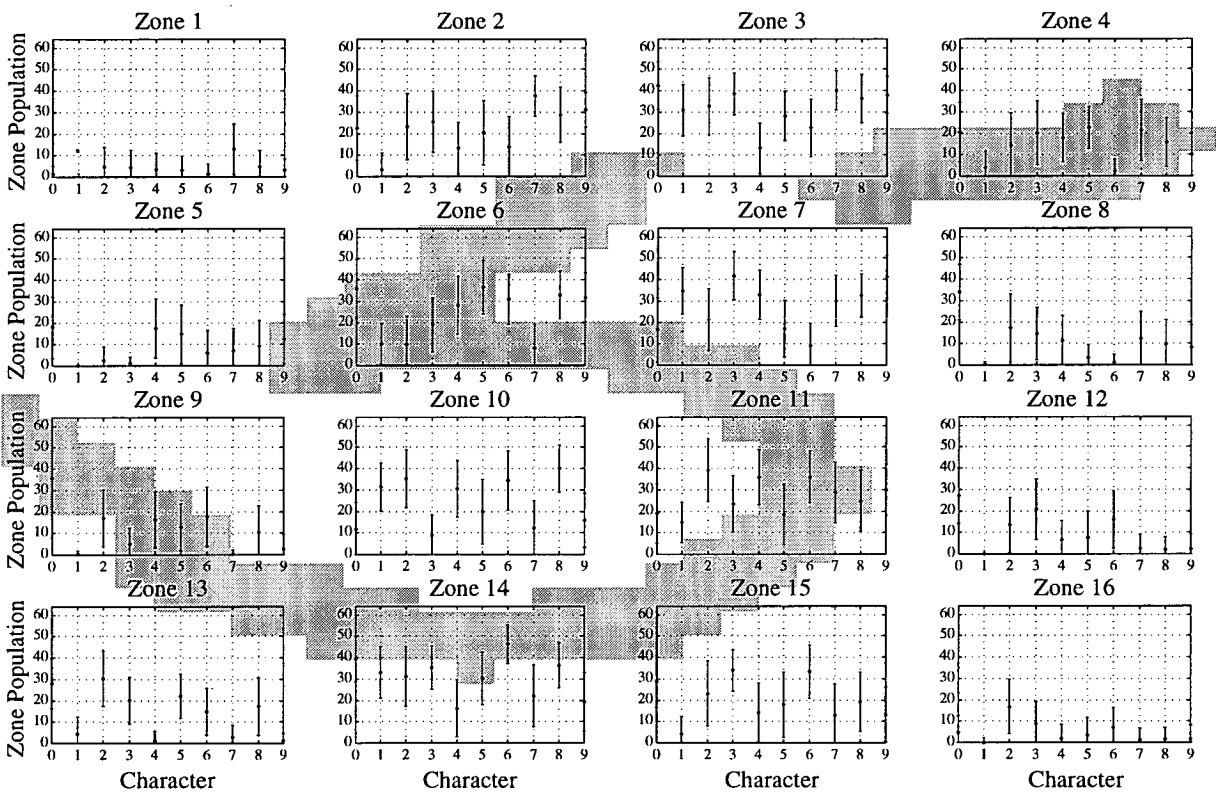
Index, $d_i$	Feature	Notes
1-3	Height/Width/Aspect ratio	Bounding box description
4	Compactness	Roundness or circularity
5	Bending energy	Object boundary complexity
6	Elongation	Stretch factor
7	Number of corners	A difficult feature to derive
8	Number of holes	Useful for character recognition
9-10	Centre of mass	Identifies position of greatest pixel mass
11	Symmetry	
12	Texture	More appropriate for remote sensing applications, as well as, segmentation
13	Population	Percentage object foreground
14	Arithmetic mean	Measure of pixel central tendency
15	Root mean square	
16	Median	Another measure of central tendency
17	Lower quartile	Lower and higher quarters of distribution
18	Upper quartile	
19	First decile	Lower and higher 10% of distribution
20	Ninth decile	
21	Variance	Measure of distribution spread
21	Absolute deviation about median	$\sum(occupancy *   f(x, y) - d_{16}  )/d_{12}$
23	Coefficient of variance	$100.0 * \sqrt{var}/d_{13}$
24	Quartile coefficient of skewness	$(d_{17} - (2 * d_{15}) + d_{16})/(d_{17} - d_{16})$
25	Percentile coefficient of skewness	$(d_{19} - (2 * d_{15}) + d_{18})/(d_{19} - d_{18})$
26	Moment coefficient of skewness	$d_{30}/\sqrt{d_{20}^3}$
27	Percentile coefficient of kurtosis	$0.5 * (d_{17} - d_{16})/(d_{19} - d_{18})$
28	Moment coefficient of kurtosis	$d_{31}/d_{20}^2$
29-30	Lowest/Highest pixel value	Contrast measure
31	3 <sup>rd</sup> moment	$\sum(occupancy * (f(x, y) - d_{13})^3)/d_{12}$
32	4 <sup>th</sup> moment	$\sum(occupancy * (f(x, y) - d_{13})^4)/d_{12}$

Table 4–1: Statistical features.

Zoning

One simple example of a linear spatial mapping is known as *zoning*. Zoning subdivides  $N \times N$  pixel images into  $M$ , non-overlapping, constant-valued,  $n \times n$  kernels that completely tile the image, such that  $Mn^2 = N^2$  and  $g_m(x, y) = 1/n^2$  [128]. This technique is also known as *pixel averaging* [13] and *coarse coding*.

Figure 4–2 shows the distribution of feature values, that were created with the NIST database, using 16 8x8 pixel zones. The results appeared to suggest that certain zones would



**Figure 4–2:** NIST: Effects of zoning upon the NIST digit database. For each image in the NIST database, the pixels values were added for zone. These summed zoned pixel features were then split into the various classes and the mean and standard deviation statistics estimated.

be better at differentiating between various classes of characters. However, without analysing the features quantitatively no comments could be made on what zones might prove the best features. The choice of a suitable subset of these features that would classify all the digits satisfactorily was a complex task.



One further problem with these zoning features was that they are highly susceptible to pixel-sized translations across boundaries. For example, vertically shifting a horizontal line of 8 pixels would generate a 16 pixel swing between vertically neighbouring zone features. One solution was to smooth the edges of the zones such that they overlapped. A simple way of performing this smoothing was to use a Gaussian kernel, instead of a constant-valued kernel, with a suitably chosen width,  $\sigma$ , as shown in Equation 4.3

$$g_m(x, y) = \alpha_m \exp \left\{ - \left[ \frac{(x - x_{0m})^2 + (y - y_{0m})^2}{\sigma^2} \right] \right\} \quad (4.3)$$

where  $\alpha_m$  is a constant.

### Projection histograms

The use of *projection histograms* was suggested in 1956 by Glauberian [45] for optical character recognition, primarily for binary images, although their use can be extended to grey level images.

The histogram features are derived by summing along parallel sections of an image, typically in either the horizontal or vertical direction. This is similar to zoning, except that one side of the zone is extended to the opposite edge of the image space. Features extracted in this way are very sensitive to rotations. Cumulative histograms, however, do tend to be less sensitive to shifts in the dominant peak of the histogram.

### Image moments

Image moments have been widely used as a source of features for classification. An excellent introduction to the use of moments for image analysis is given by Teague [122]. Moments are derived by integrating over the space of a weighted version of an input image, where the spatial distribution of the weights,  $g(x, y)$ , is controlled by the moment *order*,  $r$ . For each moment of order  $r$  there will be  $S$  coefficients,  $c_{rs}$ , related to each weighting distribution, or basis function. In general, this can be written as

$$c_{rs} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) g_{rs}(x, y) dx dy, s = 1..S. \quad (4.4)$$

If the coefficients are used as features then the discrete case is identical to Equation 4.2. The difference is in the form of the kernel. With moments the basis functions provide a quantitative method of extracting features from an image. In fact, given a sufficiently large number of ordered moments all image information will be captured, with the lower order moments capturing the more gross artifacts. For example, for geometrical, or regular, moments, of order  $r = p + q$ , where

$$g_m(x,y) = x^p y^q, \tag{4.5}$$

the first six geometrical moments can be related to physical image properties, as shown in Table 4–2.

m	r	p	q	Representation
1	0	0	0	Total image power
2	1	0	1	Image centroid in x
3	1	1	0	Image centroid in y
4	2	2	0	Size and orientation
5	2	1	1	Size and orientation
6	2	2	0	Size and orientation

**Table 4–2:** Low order regular moments.

Often, it is hoped that the basis functions are orthogonal so as to reduce any redundancy in the features. Geometrical moments have basis functions that, although complete, are not orthogonal according to the Weierstrass approximation theory [122]. Legendre basis functions, on the other hand, are orthogonal. The Legendre moment is defined by

$$g_m = \frac{(2p + 1)(2q + 1)}{4} L_p(x) L_q(y) \tag{4.6}$$

where the  $p^{th}$ -order Legendre polynomial is

$$L_p(x) = \frac{1}{2^np!} \frac{d^n}{dx^p} (x^2 - 1)^p \tag{4.7}$$

and the orthogonality is shown by

$$\int_{-1}^1 L_p(x) L_{p'}(x) = \frac{2}{2p + 1} \delta_{pp'}. \tag{4.8}$$

Both geometrical and Legendre moments were used to generate features for classification as part of this project. Another popular set of feature extraction methods that decompose images into a series of coefficients are based on unitary transforms, for example the Fourier transform.

### Unitary transforms

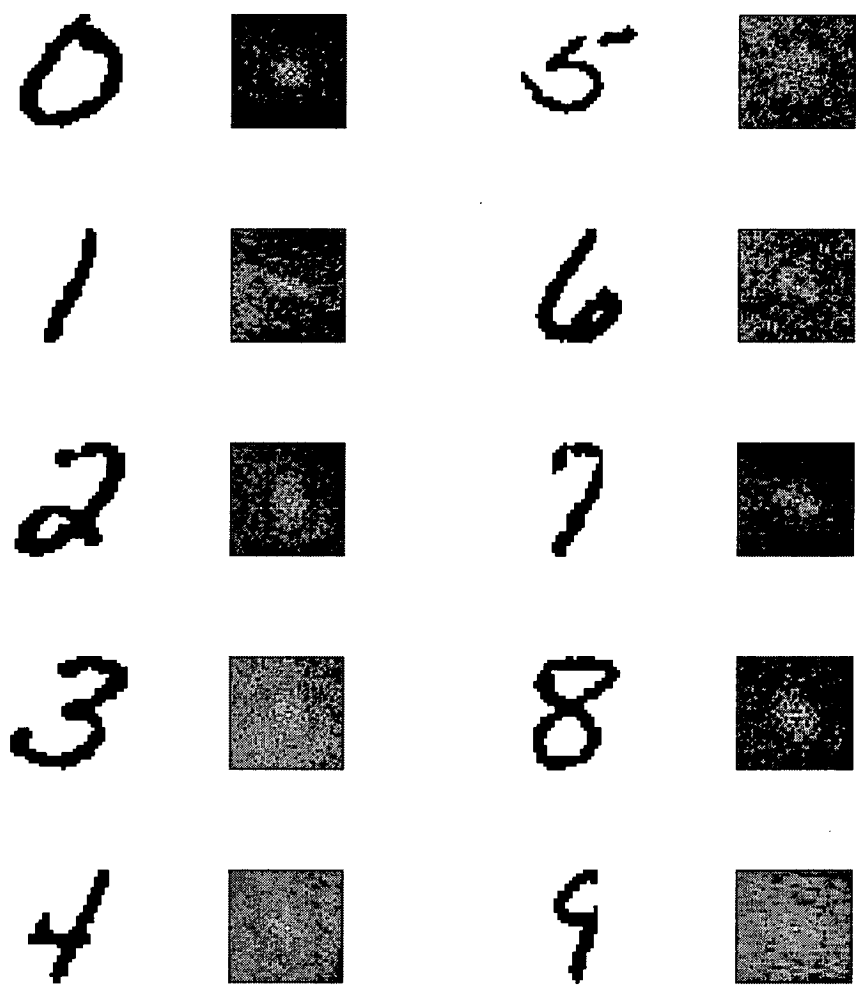
The general orthogonal series expansion of an  $N \times N$  image  $f(x, y)$  is described by the following transform pair

$$w(u, v) = \sum_x^N \sum_y^N f(x, y) g_{uv}(x, y), 1 \leq u, v \leq N \quad (4.9)$$

$$f(x, y) = \sum_u^N \sum_v^N w(u, v) g_{uv}^*(x, y), 1 \leq x, y \leq N \quad (4.10)$$

where  $g_{uv}(x, y)$  represents a set of complete, orthonormal discrete basis functions [58]. The transform coefficients can be considered directly as features, as with the image moments, or further processing can be performed on the transformed space. Some of the most popular transform basis sets are *unitary*. Returning to Equation 4.1 a unitary transform is one such that the inverse of a matrix  $\mathbf{A}$  is equal to its conjugate transpose,  $\mathbf{A}^{-1} = \mathbf{A}^{*T}$ . The Fourier transform is one example of a unitary transform, and Figure 4–3 shows the resulting transformed spaces on some digits from the NIST database. Unitary transforms have been used previously for both character recognition and ATR [3,128].

Several other types of unitary transforms that also have been used for feature extraction include the Cosine, Sine, Haar, Walsh-Hadamard, Slant and pattern transforms [3,58]. Table 4–3 gives a brief description of each of these transforms with an example of each.



**Figure 4–3:** NIST: Fourier transforms of sample digits.

With unitary transforms, as with image moments, no information is gained by the transformation. The signal energy may be compressed into a small spectral region in the transformed space but there is no assurance that features from this region will be the most important, in terms of classification.

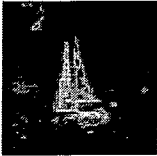
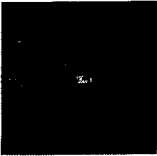
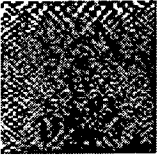
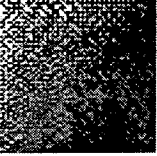


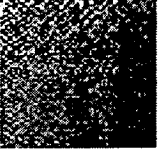
Transform	Description, $\mathcal{Z}_{u,v}\{f(x,y)\}$	Example
Pattern	One-to-one mapping of the pixel data  $f(x,y)$	
Fourier (DFT)	Asymptotically equivalent to the Karhunen-Loeve transform, the DFT is a symmetric, unitary transform with periodicity.  $\alpha \sum \sum f(x,y)[-j2\pi(ux/N + vy/N)]$	
Cosine (DCT)	This real and orthogonal transform, <i>not</i> the real part of the DFT, is often used in image compression.  $\alpha \sum \sum f(x,y)\cos[(2x+1)u\pi/2N]\cos[(2y+1)v\pi/2N]$	
Sine (DST)	A real, symmetric, and orthogonal transform. The DST is <i>not</i> the imaginary part of the DFT.  $\alpha \sum \sum f(x,y)\sin[(2x+1)u\pi/2N]\sin[(2y+1)v\pi/2N]$	
Haar	A real, orthogonal, transform, with sequence ordered basis vectors that provide a domain that is both locally and globally sensitive. The Haar transform has poor image energy compaction.	
Walsh-Hadamard	Binary transform, where $b_i(z)$ is the $i^{th}$ bit in a binary representation of $z$  $\alpha \sum \sum f(x,y)(-1)^{\sum_i [b_i(x)b_i(u)+b_i(y)b_i(v)]}$	
Slant	Defined by a recursive expression the Slant transform is real and orthogonal. It has excellent energy compaction for images.	

Table 4-3. Various unitary transforms of a sailboat ( $\alpha = constant$ ).

### Wavelet transforms

Wavelets are becoming an increasingly popular transform for pattern recognition, as well as signal representation and compression, with their ability to generate features that are both localised spatially or temporally, as well as in frequency. Classifiers based on wavelet features have been used successfully for character and speech recognition [111,60,119], breast cancer diagnosis [63], and object detection and segmentation in real IR images [125,19,21,20,133, 121].

Wavelets were originally used to analyse the temporal-frequency characteristics of non-stationary signals and introduced as a solution to the resolution problem of the short-term Fourier transform (STFT) [72]. The technique is now used often for examining image spatial-frequency content. For example, in image analysis, it is inappropriate to use Fourier transform features derived from an entire scene in order to determine the classification of a particular object, as the frequency characteristics pertaining to that particular object are spatially localised. Segmentation, as described in the previous chapter, is a crude method of localising analysis in ATR. However, it is sometimes helpful to use features that have the ability to discriminate in both frequency and space, within the confines of an object bounding box.

The continuous wavelet transform of a one-dimensional signal,  $f(x)$ , is defined as

$$\mathcal{Z}_{x_0,a}\{f(x)\} = \frac{1}{\sqrt{|a|}} \int f(x) \psi^* \left( \frac{x - x_0}{a} \right) dx \tag{4.11}$$

where  $x_0$  and  $a$  are the translation and scale parameters. The function,  $\psi$ , is known as the mother wavelet: each wavelet used at each point of the transformed space is a scaled and shifted version of this mother wavelet. The choice of the mother wavelet effects the properties of the transformation and there have been many proposed, including the eponymous Morlet, Daubechies, and Mallat transforms [72].

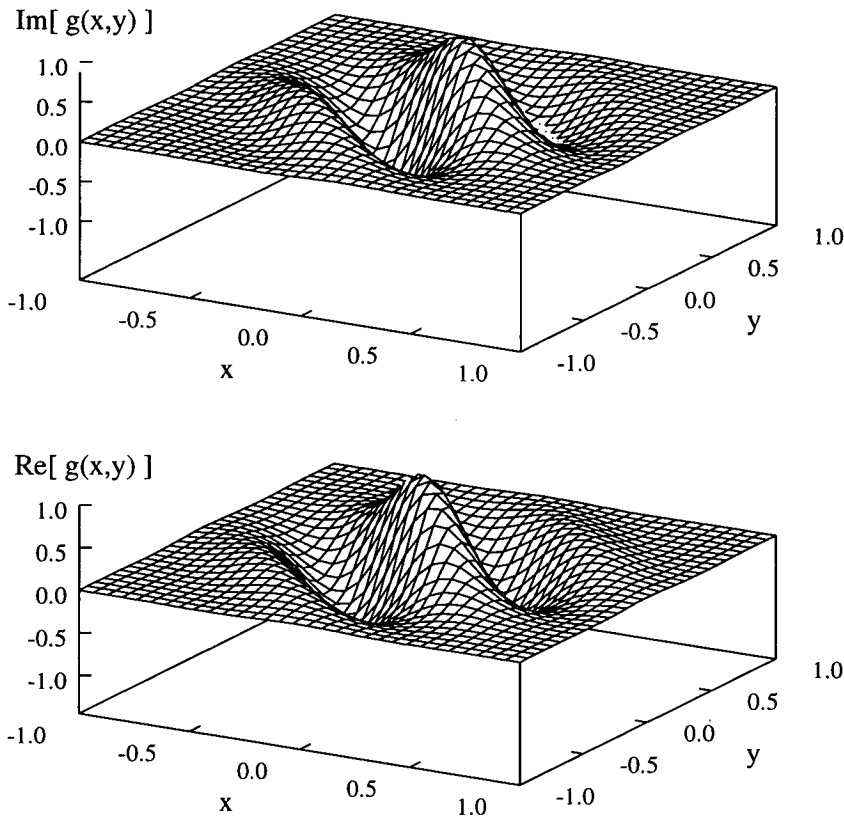
The project concentrated on the Gabor transform which is, in many ways, very similar to the wavelet expansions discussed. The Gabor transform has been used by other authors for feature extraction [84,26,133,65].

The 2-D Gabor transform consists of a Gaussian envelope centered at  $(x_0, y_0)$  with size controlling parameters  $(a, b)$ , which is modulated by a complex exponential with horizontal

and vertical spatial frequencies  $(u_0, v_0)$ . This is a linear transform, and features can be generated using Equation 4.2 with the kernel function

$$g_m(x, y; \phi_m) = \exp \left\{ - \left[ \frac{(x - x_{0m})^2}{a_m^2} + \frac{(y - y_{0m})^2}{b_m^2} \right] \right\} \cdot \exp \{ j 2 \pi (u_{0m} x + v_{0m} y) \} . \quad (4.12)$$

where  $\phi_m$  is the parameter vector  $(a_m, b_m, x_{0m}, y_{0m}, u_{0m}, v_{0m})^T$ . The transform has a spatial frequency of  $(u_0^2 + v_0^2)^{1/2}$  and a spatial orientation of  $\arctan(v_0/u_0)$ . The real and imaginary parts of one element of the Gabor transform are shown in Figure 4–4.



**Figure 4–4:** Gabor: Imaginary (*top*) and real (*bottom*) parts, where  $\phi = (0.5, 0.5, 0, 0, 1, 1)^T$ .

The elliptic generalisation of Gabor’s set of elementary one-dimensional functions [42] has many interesting properties for feature extraction, and have been noted to resemble closely the spatial-domain visual cortical filters that occur in nature [27]. The six-dimensional parameter vector  $\phi$ , given in Equation 4.12, allows for control of filter spatial orientation, frequency, spatial coverage and location. This makes the transform highly suited to feature extraction. Furthermore, these transforms achieve the best possible joint resolution in the spatial-frequency

domains, such that

$$(\Delta x)(\Delta y)(\Delta u)(\Delta v) \geq 1/16\pi^2. \quad (4.13)$$

Although sharing many wavelet properties, and being very similar to the Morlet wavelet, the general Gabor transform, is not strictly speaking a wavelet. However, its simplicity, ease-of-use, and ability to relate filter parameters to physical properties, such as orientated edges, have made the Gabor transform, as well as its real and imaginary complex parts, popular tools for image analysis. Szu *et al* provide a comparison between the wavelet and the Gabor transform, in terms of compression and recognition [118].

## Other features

These sections have only outlined some of the more popular feature extraction methods. They were chosen as features for classification in this project for their ease of calculation and popularity in the fields of ATR and character recognition. Other feature extraction techniques include the Hough, Radon, Wigner and Karhunen-Loeve (KL) transforms. The KL transform is very popular as it generates features with the largest eigenvalues, as it is hoped these features have the greatest class separability. However, it can be easily demonstrated that the good class separability is not always achieved with the KL transform, especially with real, multi-modal data. Furthermore, the KL transform requires considerable effort to compute because of the requirement to diagonalise, often very large, covariance matrices [3].

Other feature extraction techniques are based around texture measures but these often require large objects from which to derive texture cocurrency matrices [66]. Features based on describing shape by way of graphs or splines are also popular but again often not efficient to calculate [30,128].

One further subset of features, that has not yet been discussed, has a tolerance for certain deformations in the original object. These *invariant* features are very important in ATR and shall be discussed separately in Chapter 6.



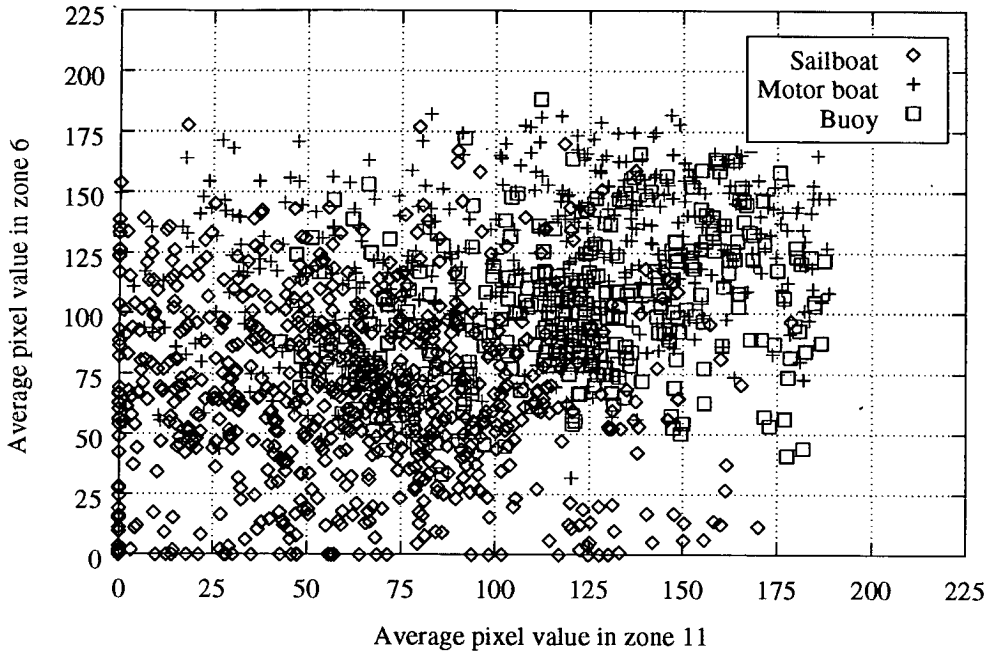
## 4.2 Feature analysis

There were several points that needed to be addressed before the application of any classification algorithm to the features. Chapter 3 showed that the distribution of seascape object width and height indicated distinct class separability. Unfortunately, much separability was discarded when scale normalisation was performed, although aspect ratio remained a potent characteriser. What was considered next was how other features separated, in fact did they separate, and if they did, were the decision boundaries likely to be linear or, in the other extreme, highly nonlinear? Both the 32x32 pixel seascape objects, and the 32x32 pixel NIST digit data, were considered.

### 4.2.1 Feature separability

Figure 4–5 shows the distribution of two zoning features with the three seascape classes. The zones relate to the zone numbers in Figure 4–2. As shown in Figure 4–5, there are indications of a degree of separability between the sailboat, and the other two classes: the motor boats and the buoys. Also, the buoy class formed two clusters, close to the motor class, allowing some nonlinear separation. However, the decision boundaries required here were by no means an indicator of the type of discrimination required with other, perhaps higher dimensional, feature spaces.

Given a decision boundary of sufficient complexity, a finite set of unique, labelled, data points can be separated exactly. However, unless the identically labelled points possess some form of cohesion, or collectiveness, no generalisation can occur. Examining the features directly, notable groupings, which provided some confidence, could be observed, as seen in Figure 4–5. Of course, it would be a considerable task to examine every possible combination of two-dimensional features and, more appropriately, impractical to view every multidimensional feature space for feature separability. There were two solutions to this problem. The first was to limit the complexity of the decision boundary and combine this with a suitable generalisation measure based only on classification scores. The second was to derive a measure of separability based directly on the features, from which an expectant generalisation could be derived.



**Figure 4–5:** Seascape: Distribution of two zoning features.

It would have been exceedingly helpful if the value of the Bayes error, the error produced by use of a Bayes decision boundary, had been known for each type of feature. However, this error was impractical to derive with the real data without explicit knowledge of  $p(\mathbf{d} \mid \omega_i)$ , but it could have been possible to place a bound on the Bayes error known as the *Chernoff bound* [90]. A much simpler bound, though, known as the *Battacharyya bound* [90], is much more widely used. These bounds, as well as other measures of conditional probability divergence such as Matusita, Patrick-Fisher, Lissack-Fu and Kolmogorov [28], are dependent on assumptions of distribution normality or availability of a mathematical expression for the distribution, or at least a reliable estimate of the probability density function (pdf) at all points. Estimating pdfs is a notoriously difficult problem. In this project a simple  $k$ -NN estimator was used, although as previously stated is not a true pdf estimator, so results using the estimate were treated with caution. Simpler measures, based on inter- and intra-class separability, that use more practical measures, required estimates of the between-class distance,  $S_b$ , and the within-class distance,  $S_w$ , defined in Equation 4.14 and Equation 4.15 respectively, where  $\mu_i$  represents the mean feature vector for all samples of class  $\omega_i$  and  $\mu$  is the mean vector of all the samples [28].

$$S_b = \sum_{i=1}^C P_i (\mu_i - \mu)(\mu_i - \mu)^T \quad (4.14)$$

$$S_w = \sum_{i=1}^C P_i \frac{1}{N_i} \sum_{k=1}^{N_i} (\mathbf{d}_k - \mu_i)(\mathbf{d}_k - \mu_i)^T$$

(4.15)

Table 4–4 outlines six separability measures that were used with both the NIST and real IR image data features before the features were classified. Measures  $J_{1-4}$  are described by Devijver and Kittler [28].

Measure	Equation	Notes
Error	$\int [1 - \max P(\omega_i   \mathbf{d})] p(\mathbf{d}) d\mathbf{d}$	The Bayes error can be estimated from the finite database using an estimate of $p(\mathbf{d}   \omega_i)$ .
Quadratic entropy, $\xi$	$\int \sum_{i=1}^C P^2(\omega_i   \mathbf{d}) p(\mathbf{d}) d\mathbf{d}$	Well known measure of information, which again requires knowledge of $p(\mathbf{d}   \omega_i)$ .
$J_1(D)$	$tr(S_w + S_b)$	
$J_2(D)$	$tr(S_b)/tr(S_w)$	
$J_3(D)$	$tr(S_w^{-1} S_b)$	
$J_4(D)$	$  S_w + S_b   /   S_w  $	

**Table 4–4:** Various separability measures.

4.2.2 3D object rotation

In Chapter 3 the problem of three-dimensional object rotation was discussed. It was suggested that the objects at different rotations existed as subclasses in a multi-modal object distribution. Figure 4–6 shows the effect a rotating ferry had on two Gabor-based features. Side on, the features existed at different positions within the motor boat distribution. Due to the effects of the mast and the narrowness of the ferry, features generated from an image of the ferry facing away were more akin to features derived from a sailboat without a sail, although in the extremes of the sailboat distribution (points C and D.) In fact, due to the mast on the ferry, it was exceedingly difficult to discriminate between it and a sailboat. This is one important area where the temporal and interpretation stages of an ATR system are so important.

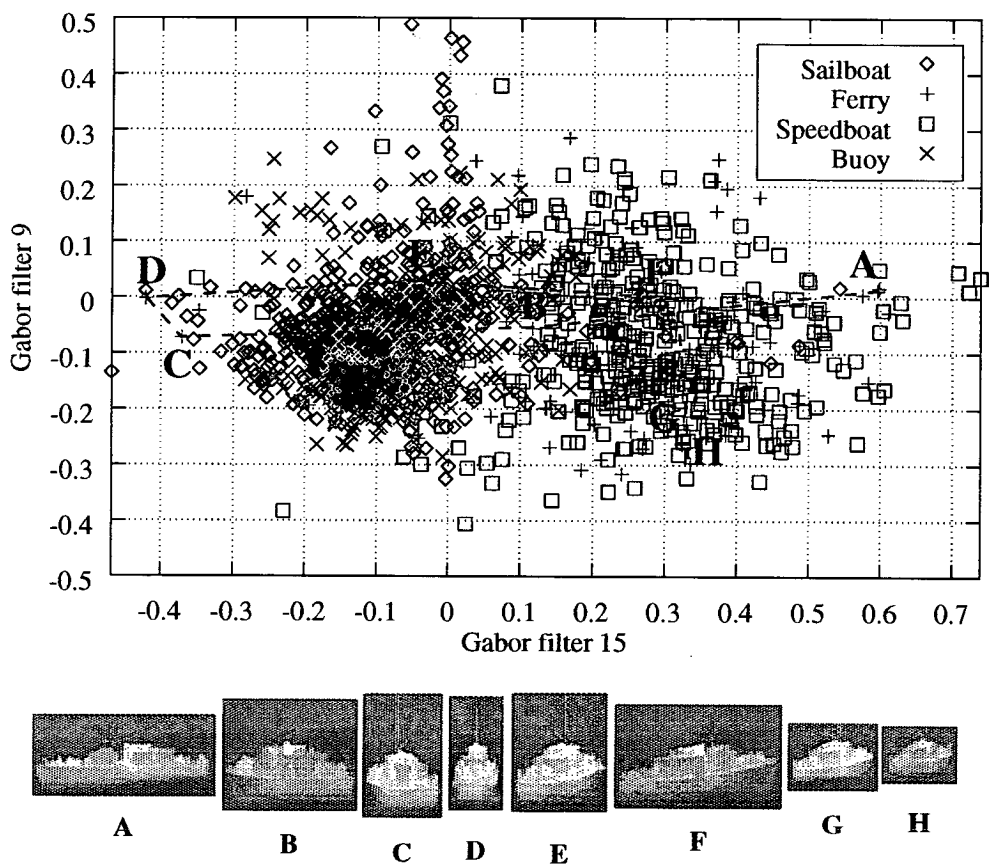


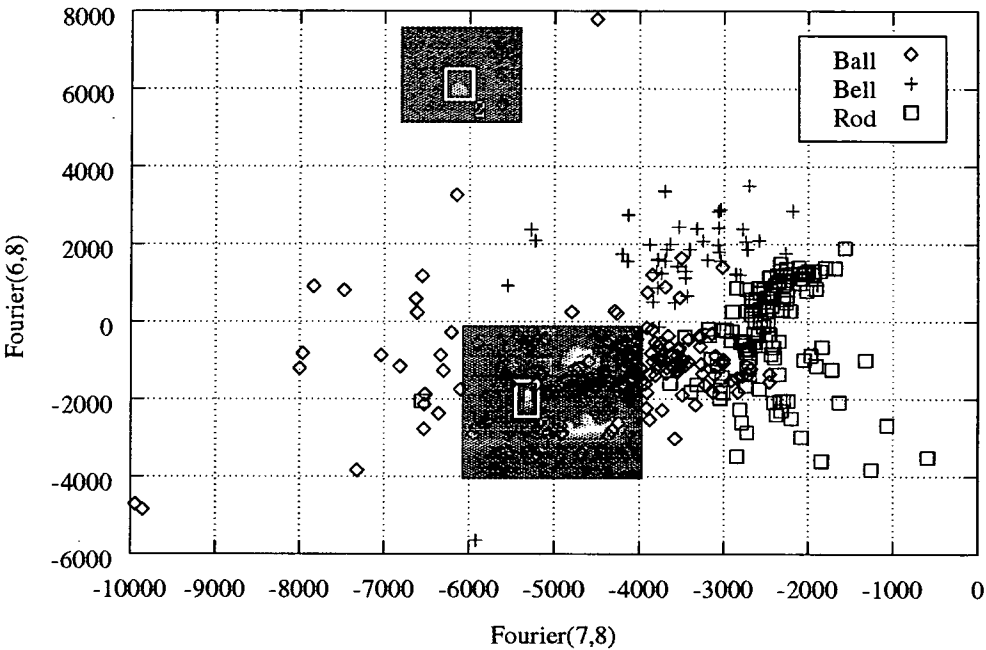
Figure 4–6: Seascape: Change in motor boat class Gabor features with rotation.

4.2.3 Outliers

Outliers are defined in this thesis as classified objects that do not belong to the class that they were assigned, or possibly even any class. They are often described as having their own distribution. In ATR outliers are termed clutter.

The process of identifying outliers with the seascape image data was simple, as shown in Chapter 3, where it was hoped that all were removed. In feature space, outliers may be spotted as isolated points well away from the main class cluster. Unfortunately, this was not always true. Figure 4–7 shows the buoy subclass distributions using two Fourier features. There appeared to be several points in the ball class that were possible outliers. Upon inspection, they were found to be slightly different from the typical ball buoy. Hence, they were in the tails of the distribution, or possibly the distribution was under-sampled with this particular, perhaps

uncommon, form of ball buoy. They were not outliers. Furthermore, there was no reason why outlier features should reside distinctly away from the distribution. For example, in ATR, clutter may have extracted properties very similar to actual objects.



**Figure 4–7:** Seascape: Fourier based features showing separability of buoy subclasses.

The manual process of corroborating the classification was assumed to have identified all possible outliers in the well-segmented database. An automated process for identifying outliers, such as clutter, is discussed in Chapter 7.

4.2.4 Multi-modality

The next issue concerned how the features were spread across the subclasses. This would effect the complexity of the discrimination boundary. As stated in Chapter 3 there existed various forms of subclasses in the seascape database. It was now appropriate to see how these translated to differences in features, if any, as some features may not have displayed any differences due, for example, to the grossness of the feature extraction algorithm. In Chapter 3 it was shown there exist object differences through design, such as with the buoy class (Figure 4–7), and also differences through object state, such as whether a sail was hoisted or in which direction the object was travelling. The latter is shown, for two statistical features, in Figure 4–8.

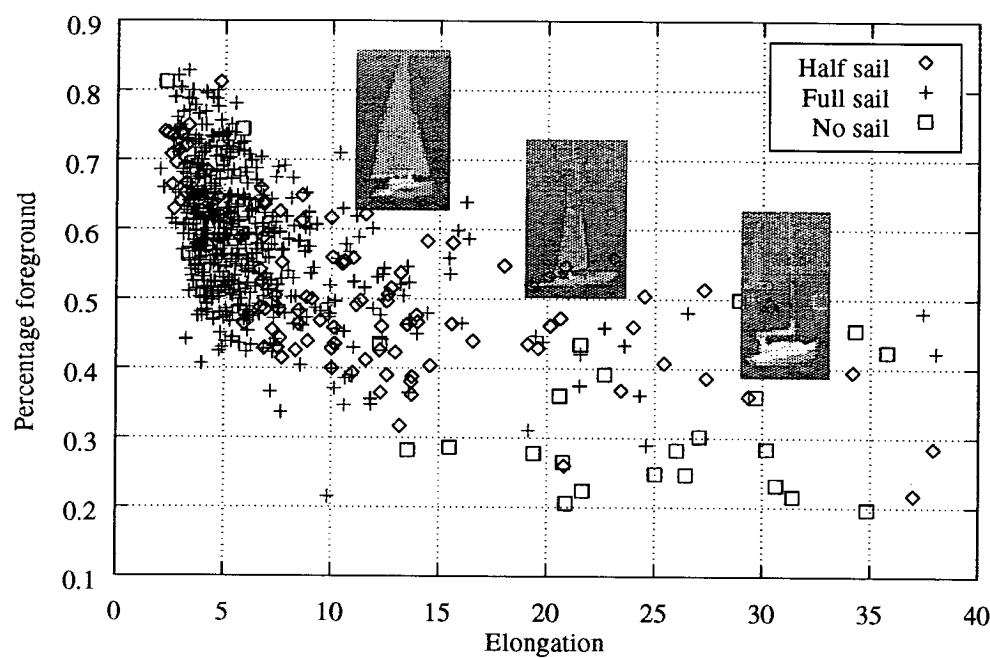


Figure 4-8: Seascape: Sailboat subclass sail states.

It was now appropriate to make this important point. If, instead of choosing sail state as the subclass, sailboat type was used, as in Figure 4-9, then no separability was visible.

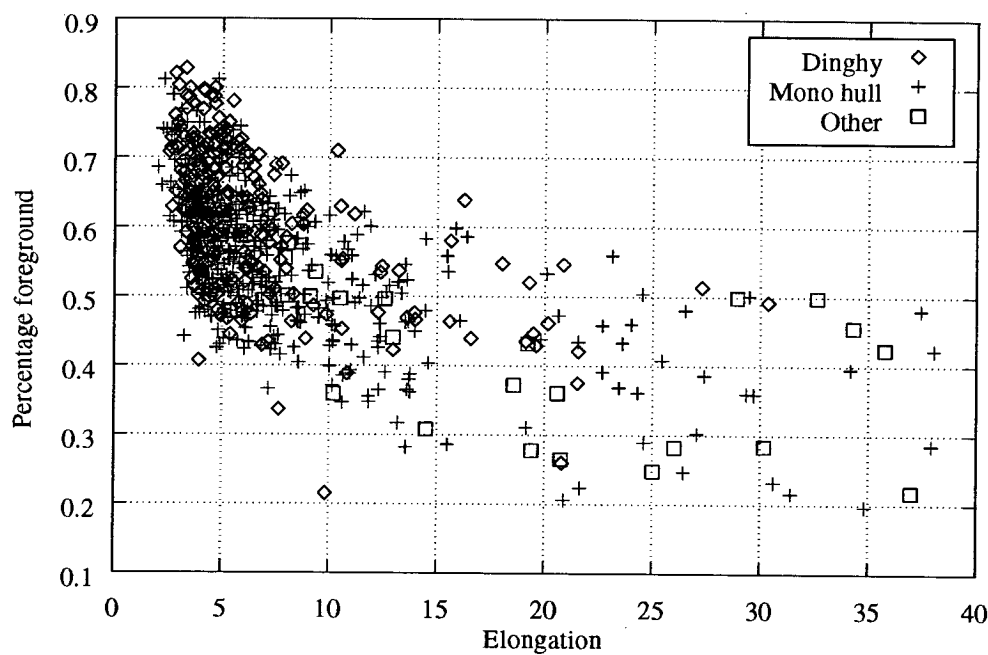


Figure 4-9: Seascape: Sailboat subclass designs.

This was easily explained as scaled sailboat types, in the same sail state, looked identical. However, the distribution in Figure 4–9 did not appear to be multi-modal. It was realised that knowledge that a distribution is comprised, of say a mixture of three Gaussians, could be very useful in deciding on a type of classifier.

It was found that subclasses did exist with some features, resulting in multi-modal class distributions. This resulted often in the need for a nonlinear classifier. However, this was not always the case, as stated earlier, classification depends on the grossness of the feature extraction algorithm. For example, buoys may have had some trait that typified them against other objects but were independent of subclass; aspect ratio, perhaps. In fact, there was often a balance between choosing features that adequately separated the main classes and features that separated the subclasses too much and added unnecessary complexity to the decision boundary.

#### 4.2.5 Feature confidence

One interesting side-issue when examining these features was how much confidence, or trust, could be placed on the actual value of a feature. That is to say, how robust were particular features to perturbations in the original image and, importantly, how confidence in these feature were affected by the original size of the image? In the project, each feature was treated with equal confidence as there was no time to examine this further. Though, it must be noted that this unlikely to be a correct assumption, for example, aspect ratio calculations will be effected less by small perturbations in object size when the object is large.

#### 4.2.6 Normalisation

In Figures 4–5 through 4–9 the features were observed having widely varying values spread across several orders of magnitude. It was therefore appropriate to normalise features such that each new feature,  $d'$ ,<sup>1</sup> had zero mean, unit variance, using

$$d'_i = \frac{d_i - \bar{d}_i}{s(d_i)}$$

---

<sup>1</sup>All features will be assumed to be normalised, hence plain  $d$  notation shall continue to be used.

where,  $\bar{d}_i$  and  $s(d_i)$ , are the  $i^{th}$  feature mean and variance respectively.

Normalisation was not necessary for MLP classifiers due to the linear scaling effect of the input layer, but it was often useful for improving rates of convergence in the network optimisation. Other classifiers though, such as  $k$ -NN, were, of course, directly effected by this type of feature scaling.

## 4.3 Preliminary classification

The features described in the previous sections all have their own characteristics and all separate classes in their own unique way. This section provides the initial classification results achieved with the two different databases, using various classifiers, and features. The features were chosen on an intuitive basis based on experience. The section starts with a look at how the classifiers, that were discussed in Chapter 2, were implemented.

### 4.3.1 Classifier experimental setup

For each of the feature types tested, eight different classifiers were used to generate comparative results. These included two generalised linear discriminants (linear and quadratic), one non-parametric classifier ( $k$ -NN), three MLP neural networks, one RBF network and one statistical classifier (MARS.) Table 4–5 lists the classifiers used with a small description of the model, and estimation method, used to determine the model parameters. These were discussed in more depth in Chapter 2.



Classifier	Notes
Linear	A generalised linear discriminant using a sum-of-squares error criterion.
Quadratic	Extension of the linear classifier to include feature product terms, $d_i d_j \quad \forall \quad i \geq j$ .
$k$ -NN	$k$ is set to 7 which were determined by trial-and-error as providing acceptable validation set errors.
MLP <sup>a</sup>	Trained using early stopping, 1-of-C output encoding, 4 hidden nodes, conjugate gradient optimisation, with a sum-of-squares error criterion, 0/1 target values, weights initially set to a random value between -0.5 and 0.5
MLP <sup>b</sup>	As MLP <sup>a</sup> but with 8 hidden nodes
MLP <sup>c</sup>	As MLP <sup>a</sup> but with 16 hidden nodes with a weight decay parameter, $\lambda$ , to control over-fitting. $\lambda$ was adjusted such as to minimise the validation set error.
RBF	Trained using supervised learning, 1-of-C output encoding, 32 hidden nodes, conjugate gradient optimisation, with a sum-of-squares error criterion, 0/1 target values, weights initially set to a random value between -0.5 and 0.5
MARS	Friedman's multivariate adaptive regression splines of degree 5 using logistic regression, with a piecewise cubic model <sup>2</sup> .

**Table 4-5:** Types of classifier implemented.

The data was randomly split into three sections, a training set, validation set and an independent test set, in the ratio 2:1:1 respectively. For example, for the three class sailboat database with high quality segmentation this was 809:400:400. The validation set was used to determine when model parameters had been suitably estimated, to avoid over-fitting. An independent test set was used to determine the actual misclassification rate. Each experiment was repeated over 10 different random splits of the data.

<sup>2</sup>FORTAN77 code courtesy of J. H. Friedman

The features used at this stage included

- Object height and width. Statistical features 1 and 2 (see Table 4–1.)
- Object characteristics. Statistical features 3,4,6,9 and 10 chosen from experience/intuition.
- Zoning of the image data.
- Symmetrical Gaussian's of width  $\sigma$  evenly spaced across the image space. A smoothed version of zoning.
- Projection histograms in both x and y directions.
- Zoning applied to various unitary transforms of the image data. Complex transform spaces are divided equally between complex magnitude and phase.
- Legendre and geometrical moments in increasing moment order.
- Centred Gabor ( $x_0 = 0, y_0 = 0$ ) based features at 5 equally spaced orientations (starting at  $0^\circ$ ) at three different spatial frequencies ( $u_0^2 + v_0^2 = 1.0, 2.25, 4.0$ ) with  $a = b = 1$ .
- Features based on object grey level distribution. Statistical features 12,13,16,17,20,25 and 27.

### 4.3.2 Classifier results

Table 4–6 provides the separability measures, given in Section 4.2.1, that were applied to the seascape 3-class database before classification. They are ordered in terms of increasing estimated Bayes error, with # representing the number of features.

The separability measure  $J_1(d)$  was unusable for assessing separability, as  $J_1 = M$ . This was due to the normalisation of the features to zero mean and unit variance. More specifically,  $J_1$  can be easily reformulated as

$$J_1 = \sum_{j=1}^M \sum_{i=1}^C P(\omega_i) \mu_{ji}^2 + P(\omega_i) \sigma_{ji}^2 = M$$

where  $\mu_{ji}$  is the mean of class  $\omega_i$  for feature  $j$  and  $\sigma_{ji}$  is the equivalent variance. However, the other separability measures appeared promising for assessing class separation, although there were several spurious looking results, for example the  $x$ -histogram features. The problem here was explained by the highly correlated nature of these features, with their immediate histogram neighbours. In fact, both  $x$ - and  $y$ -histogram features had average absolute off-diagonal cross-correlations of 0.53 and 0.44 (compare that with a Legendre value of 0.26) and more importantly had several cross-correlations greater than 0.9. It was also found that comparing feature sets of different feature dimensionality was also inappropriate.

Index	Feature	#	$E[Error]$	$E[\xi]$	$J_1$	$J_2$	$J_3$	$J_4$
1	Legendre	15	0.109	0.307	15.000	0.417	11.227	19.338
2	Zoning	16	0.112	0.321	16.000	0.511	8.474	14.381
3	Gaussian	16	0.114	0.316	16.000	0.512	9.730	17.452
4	Y histogram	16	0.126	0.356	16.000	0.693	3.796	6.153
5	Geometrical	15	0.127	0.362	15.000	0.447	10.894	18.753
6	Gaussian	9	0.142	0.398	9.000	0.610	9.782	15.439
7	Gabor	15	0.157	0.441	15.000	0.386	3.770	6.373
8	Characteristics	5	0.182	0.481	5.000	0.278	2.159	3.471
9	Width/Height	2	0.192	0.507	2.000	0.238	1.109	2.224
10	X histogram	15	0.197	0.524	16.000	0.969	7.411	8.764
11	Slant	16	0.267	0.705	16.000	0.035	0.783	1.810
12	Haar	16	0.318	0.840	16.000	0.117	1.580	2.930
13	Fourier	16	0.348	0.907	16.000	0.170	1.544	2.727
14	Grey distn.	7	0.351	0.928	7.000	0.097	0.978	2.083
15	DCT	16	0.365	0.951	16.000	0.033	0.644	1.707
16	DST	16	0.379	0.985	16.000	0.020	0.397	1.411
17	Hadamard	16	0.997	NaN <sup>3</sup>	16.000	0.001	NaN	NaN

**Table 4–6:** Seascape: Separability measures ordered by smallest estimated error.

<sup>3</sup>Could not be calculated

The actual classification results achieved are provided in terms of mean percentage correct classification in Table 4–7, and continued in Table 4–8. The values in brackets are unit standard deviation values, the results in bold represent maximum classification rate for a particular classifier, and the underlined results represent maximum classification per feature type. The original BASE classifier with 256 inputs, 16 hidden nodes and 3 outputs scored a classification rate of 88.0%.

Feature	#	Classifier							
		Linear	Quadratic	7-NN	MLP <sup>a</sup>	MLP <sup>b</sup>	MLP <sup>c</sup>	RBF	MARS
Legendre	15	85.75	89.0	<u>92.5</u>	89.25	92.0	90.0	80.5	90.75
		(1.4)	(1.1)	(0.6)	(2.1)	(1.0)	(1.5)	(5.5)	(2.2)
Zoning	16	86.0	90.0	<u>92.5</u>	90.5	92.0	91.0	<b>91.75</b>	<b>92.25</b>
		(0.6)	(1.7)	(0.6)	(1.1)	(1.1)	(1.0)	(1.5)	(1.6)
Gaussian ( $\sigma = 0.125$ )	16	86.0	<b>90.25</b>	<u>92.5</u>	<b>92.0</b>	<u>92.5</u>	<u>92.5</u>	88.0	91.0
		(1.5)	(1.4)	(0.8)	(1.2)	(1.0)	(1.0)	(3.7)	(2.3)
Y histogram	16	80.5	85.25	<u>92.0</u>	91.25	91.5	<u>92.0</u>	90.0	87.25
		(1.6)	(1.6)	(0.7)	(2.0)	(1.6)	(1.4)	(2.2)	(3.6)
Geometrical	15	<b>86.25</b>	89.0	<u>90.75</u>	90.5	<u>90.75</u>	90.5	89.25	90.25
		(1.7)	(1.7)	(1.1)	(1.3)	(1.4)	(1.1)	(1.5)	(4.0)
Gaussian ( $\sigma = 0.25$ )	9	84.0	88.0	90.25	89.25	<u>91.0</u>	89.75	89.0	90.0
		(1.5)	(1.4)	(0.8)	(1.9)	(1.1)	(1.3)	(1.6)	(1.0)
Gabor	15	80.5	89.25	89.75	89.0	<u>90.0</u>	89.75	87.0	88.75
		(1.2)	(1.4)	(1.6)	(1.6)	(1.2)	(1.5)	(2.2)	(1.2)
Characteristics	5	69.5	76.0	85.5	88.0	88.0	87.75	86.0	<u>89.25</u>
		(1.9)	(1.9)	(1.4)	(0.9)	(1.6)	(2.1)	(1.5)	(1.5)
Width/Height	2	69.5	72.0	82.0	81.25	81.5	<u>82.5</u>	<u>82.5</u>	81.5
		(2.1)	(1.9)	(1.6)	(2.3)	(2.0)	(2.0)	(1.5)	(1.7)

**Table 4–7:** Seascape: Classification results. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests. The values in bold represent highest mean classification for a type of classifier and the values underlined the highest mean classification for a type of feature.

Feature	#	Classifier							
		Linear	Quadratic	7-NN	MLP <sup>a</sup>	MLP <sup>b</sup>	MLP <sup>c</sup>	RBF	MARS
X histogram	16	71.75	77.0	<u>86.25</u>	83.75	85.5	85.0	81.5	85.5
		(1.9)	(2.1)	(2.0)	(1.4)	(0.9)	(2.2)	(3.6)	(2.4)
Slant zoning	16	64.25	64.0	74.5	73.5	<u>75.75</u>	75.5	72.5	75.5
		(2.1)	(3.2)	(1.5)	(2.0)	(1.9)	(1.7)	(3.2)	(2.5)
Haar zoning	16	72.0	72.25	74.5	73.5	74.25	73.75	<u>77.5</u>	76.5
		(2.1)	(1.8)	(1.3)	(1.9)	(1.5)	(1.4)	(2.4)	(2.3)
Fourier zoning	16	64.0	70.0	73.5	74.5	73.0	74.5	72.75	<u>76.0</u>
		(1.6)	(1.3)	(2.2)	(2.7)	(2.9)	(2.4)	(2.6)	(2.8)
Grey dist.	7	69.25	70.75	72.25	73.5	<u>74.75</u>	74.0	71.75	73.5
		(2.5)	(1.4)	(2.6)	(3.0)	(2.9)	(2.6)	(3.8)	(2.9)
DCT zoning	16	63.5	68.75	<u>74.5</u>	72.25	73.75	73.0	67.25	74.5
		(2.2)	(2.9)	(1.3)	(1.6)	(3.8)	(1.7)	(5.5)	(2.5)
DST zoning	16	59.0	66.25	70.25	64.75	65.0	65.0	<u>71.25</u>	70.0
		(1.7)	(3.7)	(1.5)	(3.1)	(3.7)	(2.6)	(2.3)	(2.4)
Hadamard zoning	16	20.75	23.25	<u>46.25</u>	<u>46.25</u>	<u>46.25</u>	44.25	<u>46.25</u>	46.0
		(2.2)	(2.6)	(2.2)	(2.2)	(2.2)	(2.3)	(2.2)	(2.2)

Table 4–8: Seascape: Classification results (continued).

Table 4–9 provides the NIST classification results based on a subset of the seascape features. Features that have been derived explicitly for digit and character recognition were not tested, as only comparative results were required. Also, some features, such as those derived from the grey level distribution, were not suitable for this database and were excluded.

The zoning features and moments perform particularly well, whilst the binned transforms results were poor <sup>4</sup>. The Y histogram data results, with low linear results, coupled with high result variance, indicated collinearity in the data. In fact, it was found that each bin in the histogram was highly correlated with its neighbour. Subsequently, every other bin was used as a feature with significantly better results and half the number of features.

<sup>4</sup>Other poor results omitted

Feature	#	Classifier							
		Linear	Quadratic	7-NN	MLP <sup>a</sup>	MLP <sup>b</sup>	MLP <sup>c</sup>	RBF	MARS
Gaussian	9	69.0	76.75	<u>83.5</u>	58.75	75.5	78.5	77.5	78.0
( $\sigma = 0.25$ )		(1.0)	(1.2)	(0.9)	(3.8)	(2.3)	(2.0)	(2.3)	(2.1)
Gaussian	16	76.25	<u>89.75</u>	89.5	61.0	81.0	83.0	82.75	81.75
( $\sigma = 0.125$ )		(1.5)	(0.8)	(1.0)	(3.5)	(1.3)	(1.4)	(1.0)	(1.5)
Y histogram	16	18.0	53.75	<u>57.5</u>	-	-	-	-	-
		(8.9)	(4.0)	(4.1)	-	-	-	-	-
Y histogram	8	55.25	<u>81.75</u>	78.0	54.25	71.0	75.0	75.5	75.25
		(1.9)	(1.3)	(1.7)	(0.9)	(1.8)	(0.9)	(1.4)	(1.1)
X histogram	16	36.0	<u>53.25</u>	-	-	-	-	-	-
		(1.5)	(1.9)	-	-	-	-	-	-
Zoning	16	<b>82.75</b>	<b><u>92.5</u></b>	90.5	<b>62.5</b>	83.5	86.25	86.25	85.75
		(1.4)	(1.0)	(1.1)	(3.1)	(0.9)	(1.1)	(1.3)	(1.4)
Fourier zoning	16	43.25	<u>45.0</u>	-	-	-	-	-	-
		(1.4)	(1.1)	-	-	-	-	-	-
DCT zoning	16	24.75	<u>28.5</u>	-	-	-	-	-	-
		(2.0)	(1.8)	-	-	-	-	-	-
Legendre	15	80.25	90.0	<u>90.5</u>	59.75	78.5	80.75	80.0	80.5
		(1.6)	(1.0)	(1.0)	(4.7)	(1.4)	(1.2)	(1.0)	(1.2)
Geometrical	15	81.75	81.75	<b><u>92.75</u></b>	54.75	<b>84.25</b>	<b>87.0</b>	<b>88.0</b>	<b>87.5</b>
		(1.8)	(1.1)	(0.8)	(4.8)	(0.9)	(0.9)	(1.4)	(1.3)
Gabor	15	60.5	76.5	<u>79.75</u>	45.5	63.0	65.0	65.25	66.0
		(1.7)	(1.1)	(1.4)	(3.2)	(1.3)	(1.2)	(1.1)	(1.6)

**Table 4–9:** NIST: Classification results. Each score is the mean percentage classification over 10 different samples each consisting of 800 test vectors. The value in parentheses is the standard deviation over the 10 tests. The values in bold represent highest mean classification for a type of classifier and the values underlined the highest mean classification for a type of feature.

### 4.3.3 Comments

Several conclusions were drawn from these initial experiments:

- The features chosen all required some form of nonlinear discrimination to achieve an acceptable classification rate, of approximately 90% with the best features derived from the seascape data. A nonlinear solution was required.
- For the seascape data the best classification results were 5% better than the original BASE system employed, with a network with only 5 inputs equalling the original classifier performance.
- Feature separability measures indicated which of a group of features would provide good classification. They do not appear to be good enough to predict accurately which particular feature would be superior.
- In some cases a very simple quadratic classifier sufficed, with only 1-2% classification loss.
- Different classifiers found it easier to discriminate between different types of feature. For example, with the seascape data the linear classifier produced its best results with the geometrical features.
- The NIST classifier required far more complexity in the nonlinear classifiers to separate the higher number of classes than the seascape data.
- A greater number of features did not always produce better classification results.
- Zoning with relatively large kernels worked well in the spatial domain, but poorly in the frequency domain.
- Different features tended to work better with different databases. So for each new database the right type of feature had to be found.

Of the actual features themselves, zoning of the image data generated better results than any zoning of the Fourier, DCT, and other unitary transforms. This should have been expected as many of these transforms, by their nature, store the majority of their energy in a small region of the transformed space. This meant that most of the features contained, typically, little information. It may be more appropriate, in these spaces, to select individual points in the space; for example, the point representing the overall object mean luminance. However,

for a 32x32 pixel image there are 1024 possible features and choosing the right subset is very difficult. The next section examines methods that were employed for automatically selecting the number and type of features for successful classification.

One final set of experiments that were performed repeated the previous tests but with 16x16 pixel objects. It was found that a 1-2% drop in classification was the penalty for reducing the image size by 4. No improvement was noticed when 64x64 images were tested. This justified using the 32x32 pixel size for the objects.

## 4.4 Feature Selection

As seen in the previous section, examination of simple separability measures indicated which features would provide good classification rates, and could have avoided the estimation of many of the classifier models. Unfortunately, these measures were unable to rank the importance of selected subsets of features without actually testing every subset. In this section the problem of feature selection is explored further. Two points must be reiterated: features that provided good class separability with one database were not necessarily as successful with another database; adding more features did not necessarily improve classification. In fact, the problem was now to find a minimal-sized set of features that provided an adequate misclassification rate for each of the tasks at hand.

Consider  $\mathbf{d}'$  to be the vector of all available features, from whatever source. Furthermore, let  $\mathbf{d} \subset \mathbf{d}'$ , be the set of  $M$  features to used to classify the object. The aim of feature selection then is to find the vector  $\mathbf{d}$  such that  $\mathbf{d}$  maximises a user classification criterion function,  $J(\cdot) \forall \mathbf{d}$ , whilst minimising  $M$  at the same time.

### 4.4.1 Using *a priori* knowledge

The most obvious way of choosing a set of features was to use *a priori* knowledge to guide feature selection. For example, knowing that sailboats were thin and tall, whilst motor boats were wide and short would indicate height and width as a good feature. This was shown to be correct in the previous section. Another example, would be to choose particular elements of a



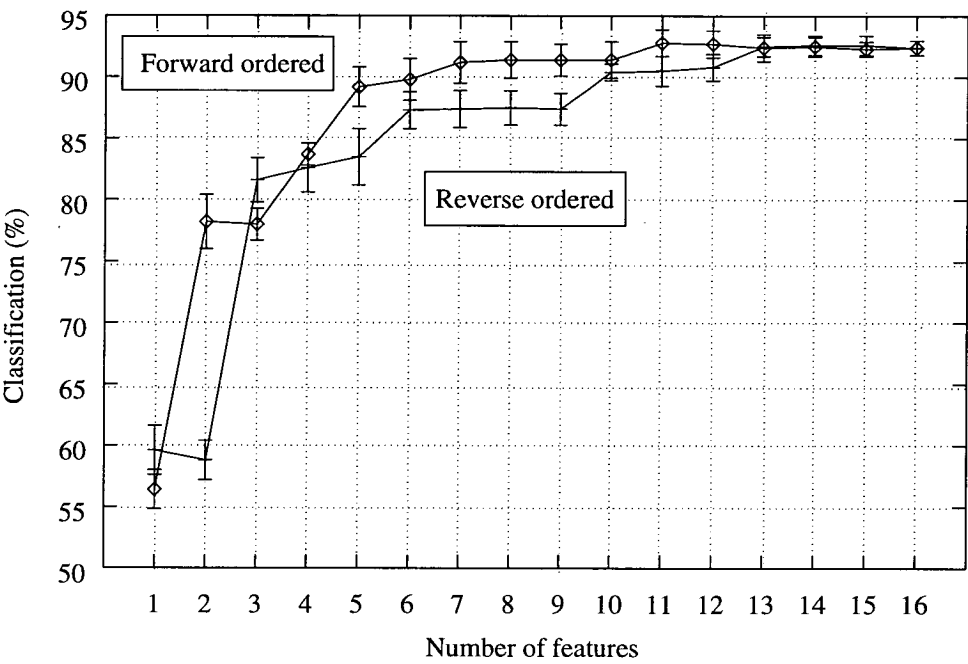
zoning feature set by examining the separation of classes in each zone, such as with the NIST data in Figure 4–2. This implied rating features individually, ignoring feature relationships.

The zone features from the NIST data were ranked according in order of perceived increasing discriminatorial power. This formed the ordered feature set

$$\{Zones : 13, 9, 10, 6, 7, 3, 2, 8, 12, 4, 14, 16, 5, 15, 1, 11\}.$$

(4.16)

Figure 4–10 shows how the classification rate changed as features were added to the number of classifier inputs in forward, and reverse, order. The error bars represent  $\pm 1$  standard deviation.



**Figure 4–10:** NIST: Increasing number of intuitive features.

Better results were achieved with forward ordered data with the classification rate reaching within 1.5% of the maximum value at 8 features. Note that the maximum rate was achieved with 11 features, rather than 16.

4.4.2 Individual feature selection

In this section, the possibility of individually rating each feature, is discussed. Individual rating was attractive for determining which unitary transform features, discussed earlier, would be

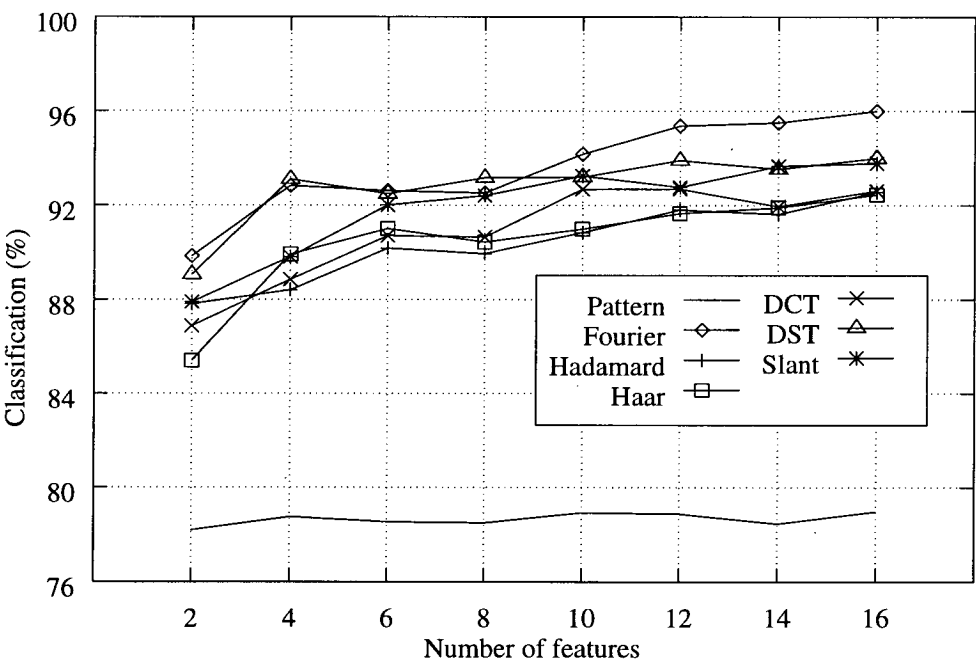
appropriate. This approach is also known as the *"Method of Best Features"* [28]. In this case, Wilks'  $\Lambda$  statistic was used to rate each feature [52].

Wilks'  $\Lambda$  statistic is simply the reciprocal of the separation measure  $J_4$ . This can be simplified to consider the separation of individual features and was applied to the unitary transform data, which previously classified poorly when features were binned. To calculate Wilks'  $\Lambda$  required calculation of transform coefficient (feature) variance. This was simple for non-complex data, but for Fourier data, the variance was calculated as

$$\sigma^2_{\nu}(k,l) = E \left[ \left| \nu(k,l) - \mu_{\nu}(k,l) \right|^2 \right]$$

where  $\nu(k,l)$  are the transform coefficients [46]. For each complex transform coefficient two features were generated.

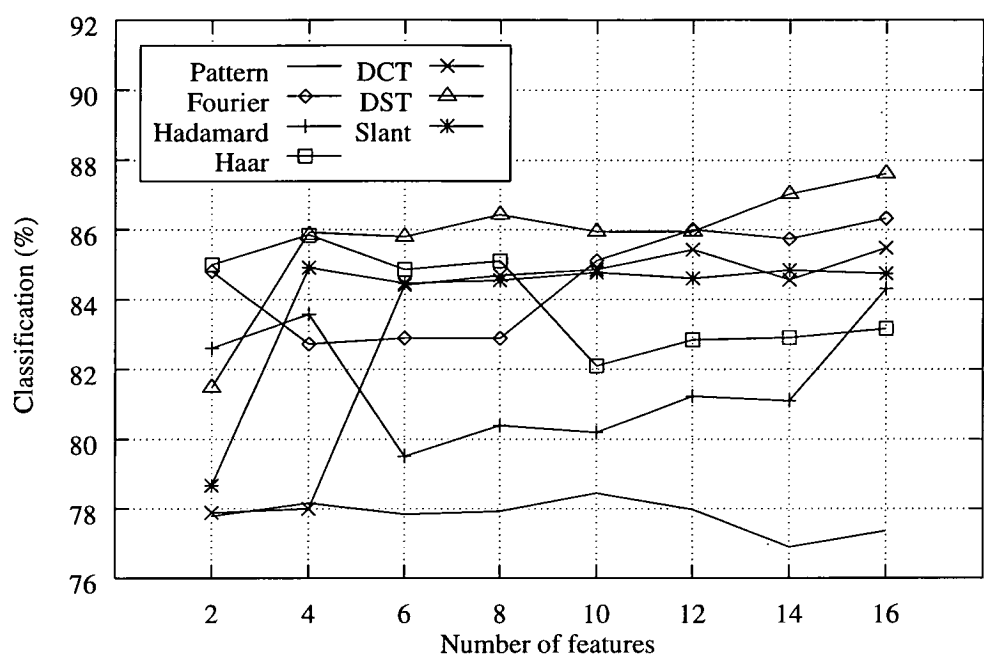
Figure 4–11 and Figure 4–12 <sup>5</sup> shows a significant improvement in seascape classification



**Figure 4–11:** Seascape: 7-NN results for transform features chosen by Wilks'  $\Lambda$ .

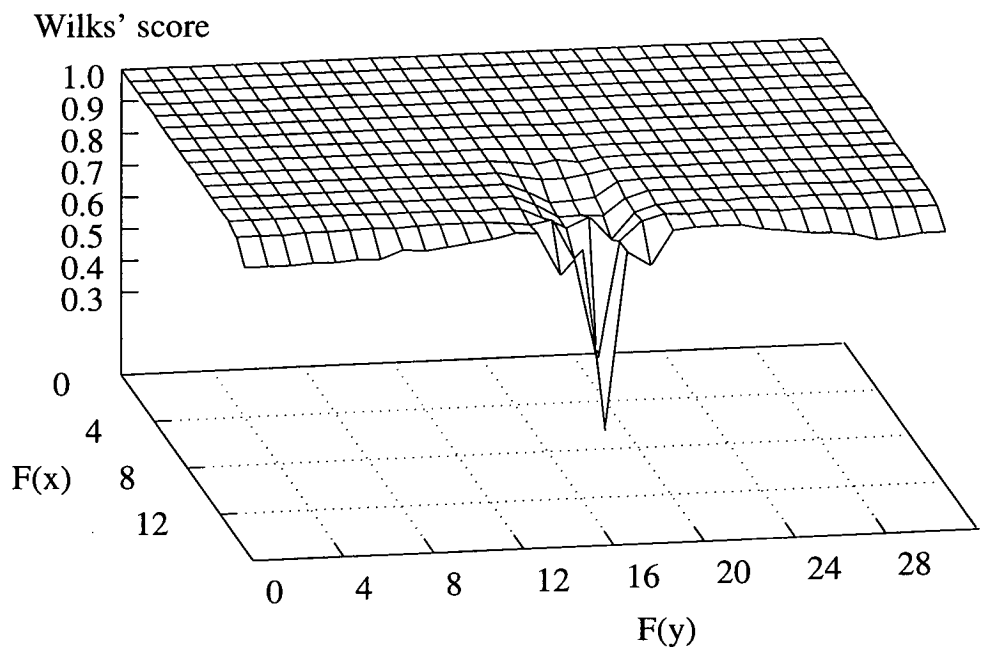
using this method with these features, especially the Fourier features providing the best results yet of 96.0%. The exception was the pattern space result. The pixel features were derived

<sup>5</sup>Error bars are not shown for clarity purposes.



**Figure 4-12:** Seascape: Linear classifier results for transform features chosen by Wilks'  $\Lambda$ .

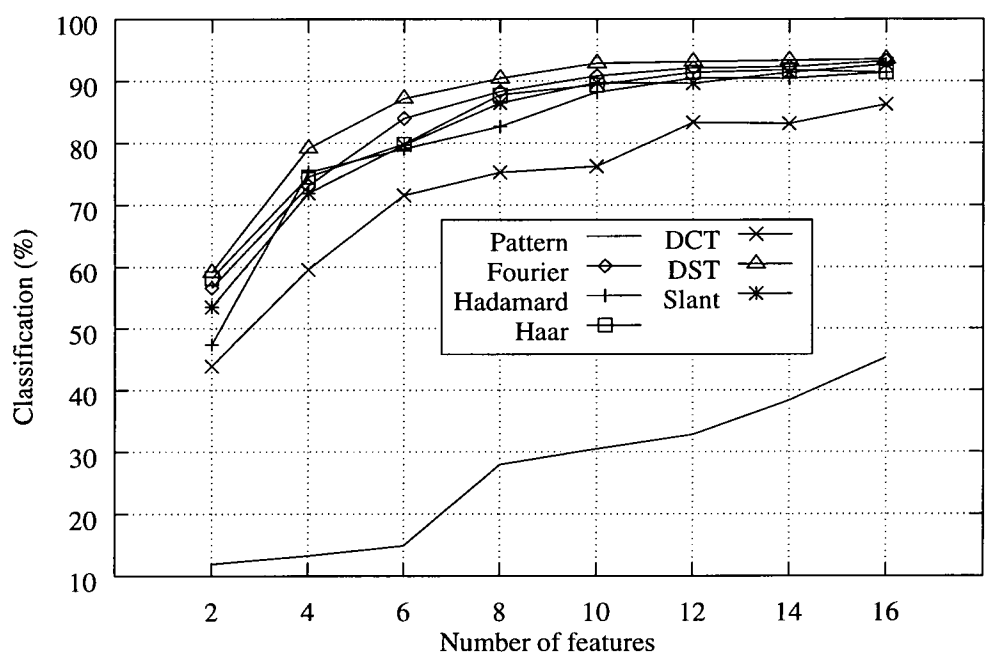
using selected zoning features whereby the size of each zone was one pixel in area. Compare this with the large kernel zoning success and failure with pattern and frequency-based features given previously.



**Figure 4-13:** Seascape: Low Wilks' score indicates good, in this case, Fourier features.

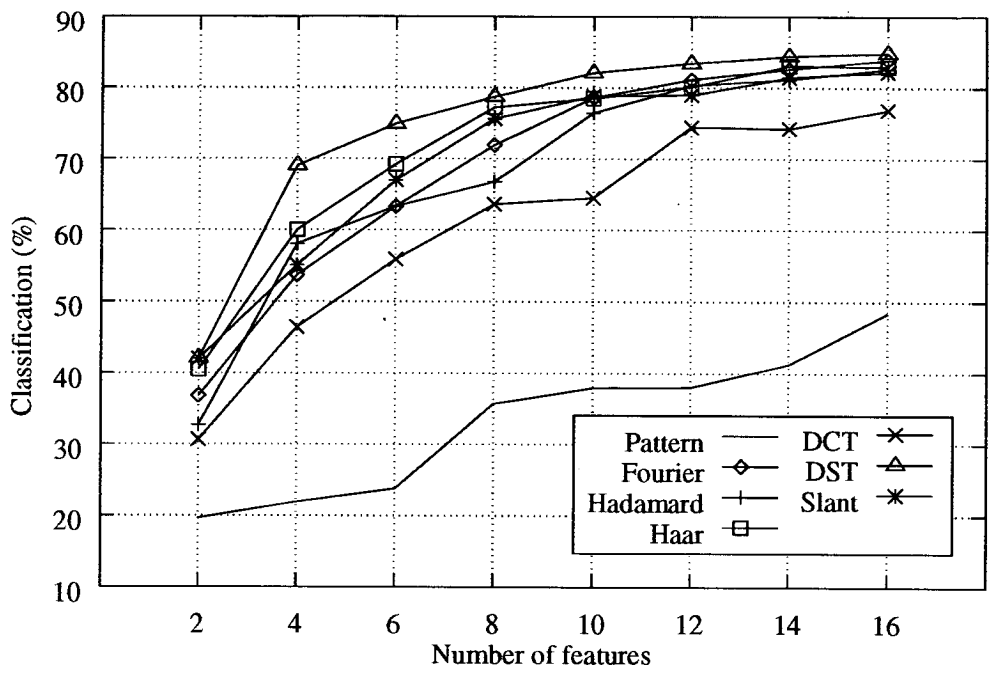
The features chosen by Wilks'  $\Lambda$  were predominately low frequency components, as shown in Figure 4–13, where the value of the Wilks'  $\Lambda$  for the Fourier transform is shown for all frequencies.

Figure 4–14 and Figure 4–15 demonstrate similar results with the NIST digit data. In this case however, the DST features outperformed what was previously the best features, the Fourier features.



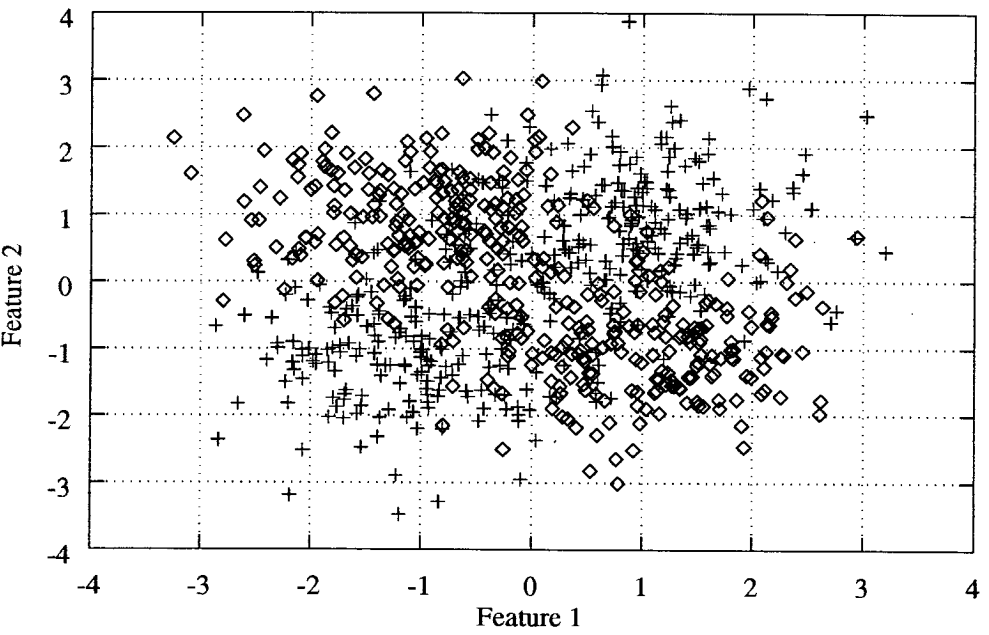
**Figure 4–14:** NIST: 7-NN classifier results for transform features chosen by Wilks'  $\Lambda$ .

This was an improvement, but what was lost by considering the features individually? In some cases, where there was feature correlation, for example, or the existence of subclasses, this single feature approach would fail. This latter problem is demonstrated in Figure 4–16 where the two classes are separable in two dimensions but, when considering each feature individually, the opposite is true.



**Figure 4–15:** NIST: Linear classifier results for transform features chosen by Wilks'  $\Lambda$ .

The solution would be to apply Wilks'  $\Lambda$  to the entire feature space. For small dimensional data this would work well, but for more much higher dimensional feature spaces this statistic became less practical.



**Figure 4–16:** The multi-modality problem with individual feature selection.

The next section examines some techniques that were implemented for choosing  $M$  features from  $N$  when the features were not treated separately.

4.4.3 Subset selection

The simplest method of determining which subset of  $M$  features from a set of  $N$  features will be optimal for classification is that of *subset selection*. In this approach an estimate of the true classification is determined for each combination of  $M$  from  $N$ . This will require  $N!/(N - M)!M!$  estimates. For small  $N$  this technique is ideal but as Figure 4–17 shows the number of estimates required soon exceeds any level of practicality with any moderately sized database.

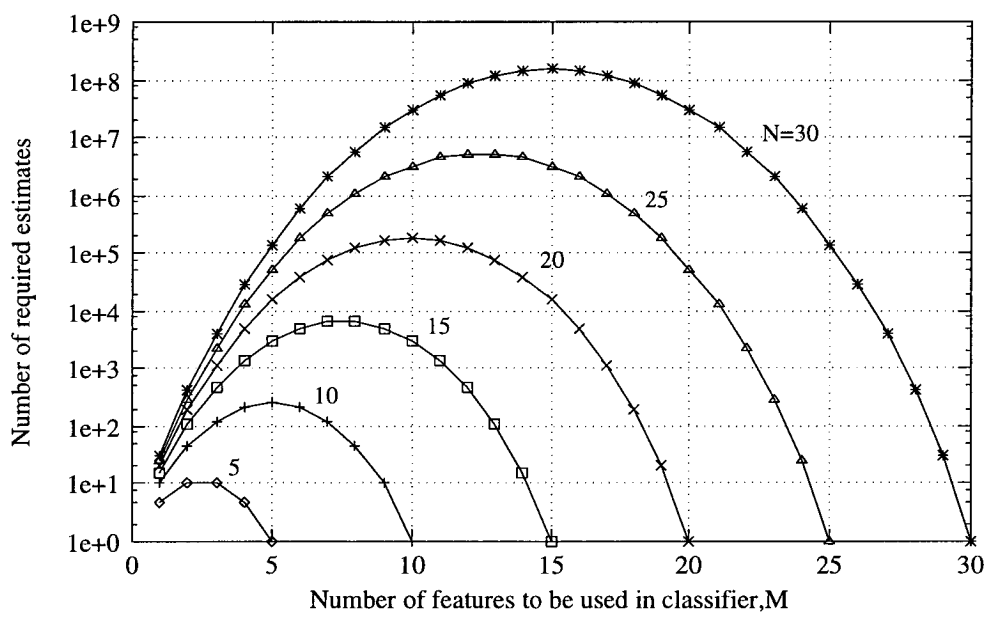


Figure 4–17: Increasing the number of available features.

This is known as an *exhaustive* search. There are two other types of search algorithm, *optimal*, and *suboptimal* [28]. Suboptimal searches, such as individual feature selection with Wilks'  $\Lambda$ , generalised sequential forward (or backward) selection and "Plus  $l$  Take away  $r$ " often lead to suboptimal features. Optimal searches, such as Branch and Bound (BaB), implicitly inspect all  $d$  out of  $D$  possible subsets, without requiring an exhaustive search. It uses a top down search procedure using a feature set tree, which allows for backtracking to counter problems

#	Index	Time (seconds)	Linear (%)	7-NN (%)
2	14 38	5	73.75	81.5
4	25 37 38 39	212	80.75	86.0
6	3 13 32 37 38 39	4206	81.75	88.75
8	3 5 13 20 32 37 38 39	28139	83.5	91.25

**Table 4–10.** Seascape: Gabor features chosen using branch and bound algorithm. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors.

of combinations of features. Both types of searches require a criterion, such as  $J_4$ , to direct the search path.

Equation 4.17 shows the set of eight features chosen by the BaB algorithm on the 16 zoning features derived from the NIST data. There are many similarities between these eight features and the eight intuitive features given in Equation 4.16. In classification tests there were no differences in generalisation.

$$\{BaB : 2, 6, 7, 9, 10, 13, 14, 15\}$$
$$\{Intuition : 2, 3, 6, 7, 8, 9, 10, 13\}$$

(4.17)

Next a set of 40 Gabor-based features from the seascape data were derived. These represented a greater spread of orientations, frequencies, and filter centres than previously used. The number of features used, feature index, time to perform BaB algorithm<sup>6</sup> and classification results are recorded in Table 4–10. There was an improvement with just 8 features using the BaB selected Gabor features. However, the time required to calculate the larger set of features soon became impractical. Also, note that the features chosen did not remain constant as the number of features were increased. This is a good example of the relationship between features and their effect on classification performance. The actual Gabor features, given by their index, relate to higher frequency, off-centre, filters.

One further experiment demonstrated that this method was not infallible. The 32 seascape

<sup>6</sup>Timed on a 167MHz Ultraspac I

statistical features, given in Table 4–1, were input to the BaB algorithm. The following 5 features were selected: Height, Width, Population, Ninth decile, and the Third Moment. This produced a classification rate 5.0% less than when the 5 intuitive statistical features were chosen. Even when 16 BaB selected features were classified the classification rate was worse. Furthermore, the 5 intuitive features were included in this set of 16! More is not always better.

It has been shown in this subsection that improved classification can be achieved through careful feature selection. However, these techniques used had several problems as shown in the box below.

- There was no consideration of the original pattern space. The selection techniques only selected the best out of the features provided.
- The process of feature selection could be laborious, especially as the size of the original feature set increased.
- Selection techniques often make assumptions concerning the underlying distributions of the features that with real, multimodal, data sets are often false.
- Estimating the most suitable features is often not performed with respect to the classification error criterion, which ultimately dictates classification performance.
- Feature selection techniques tend to work well only with reasonably sized feature spaces to begin with, which may not span the whole range of interesting possible features.
- The feature selection criterion may attempt to give features that would more suited to a less powerful classifier, especially a linear classifier.

Before analysing some of the classification results in more detail two other commonly used feature selection techniques shall be discussed.



4.4.4 Reconstruction

A very popular, and misguided, method for determining the number of features with image databases is that of signal reconstruction. This approach is common with features based on image moments, and unitary transforms, whereby the result of an inverse transform, based on truncated series of ordered coefficients, is compared with an original. The difference is a measure of the amount of information, recorded within the finite series of coefficients, or features. This measure is correct but it is a measure of *representational* information, and **not** *discriminational* information. This is best described with the aid of an example. Consider the task of discriminating between a triangle and square. The discriminational information is completely encoded in the number of vertices. All other information, such as length of edges, edge thickness and colour, for example, are superfluous.

Figure 4–18 plots 7-NN classification rate against a number of geometrical moment features. The features were determined by either increasing moment order (up to order 5, giving 15 features) or by BaB selection from an original set of 21 features, including all moments, up to and including order 6. Only the first 10 results for the latter are shown but it is clear that the same classification rate was achieved with far less features using careful feature selection.

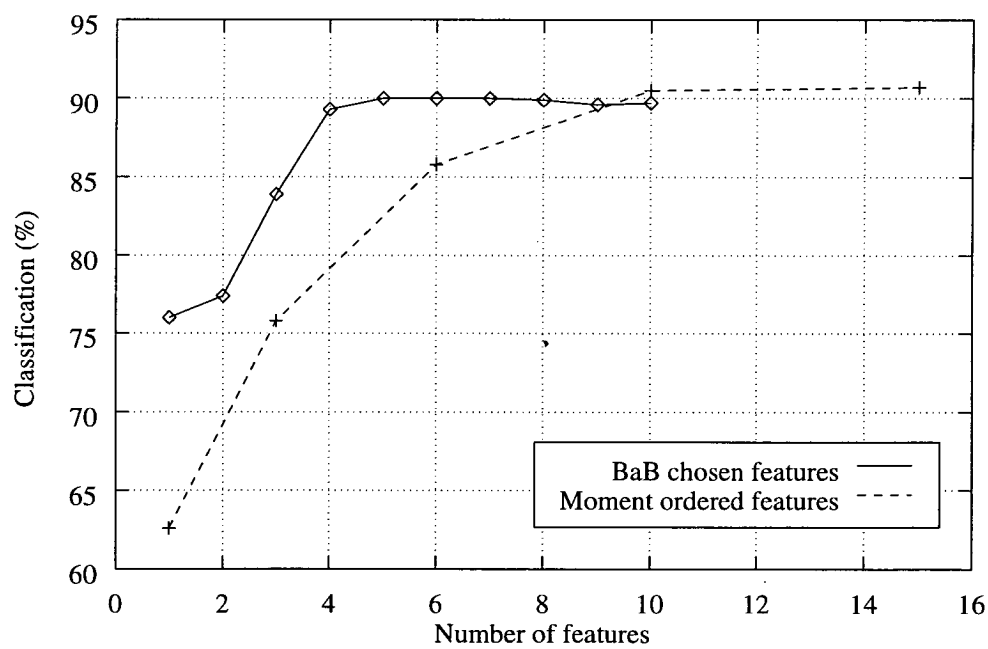


Figure 4–18: Seascape: 7-NN classifier results for geometrical moment features.

4.4.5 Other feature selection techniques

Saliency is a further technique that has been used in feature selection with neural networks. Saliency uses a trained neural network model to determine the contribution of each input feature to the final classification score. This has two major disadvantages: the user is restricted in the number of features that can be tested in order to constrain the model to a manageable size; and the technique requires the repetitive training of the neural network model.

Another method for detecting the relevant components of the feature vector is the Automatic Relevance Detection method of MacKay and Neal [74]. This is a fully Bayesian approach successfully employed by Williams and Vivarelli for classifying segmented images [130].

4.5 Analysis

This penultimate section examines why particular features performed better than others for certain databases and why particular objects were repeatedly misclassified. A first step was to examine the confusion matrices, of which six are shown for the seascape data in Tables 4–11, 4–12, and 4–13. This shows the near perfect *linear* separability of the sailboat and motor classes with very simple features. This was also seen with all the other features, using the image-size normalised object data. This suggested that a nonlinear solution was not required to separate these two classes so the significant improvements in classification rate with the nonlinear classifiers must have been due to the additional buoy class.

Guess	Correct class				Total
	Sail	Motor	Buoy		
Sail	153	2	<b>46</b>		201
Motor	2	119	4		125
Buoy	<b>25</b>	2	47		74
Total	180	123	97		400

Guess	Correct class				Total
	Sail	Motor	Buoy		
Sail	90	0	<b>12</b>		102
Motor	1	105	0		106
Buoy	<b>89</b>	<b>18</b>	85		192
Total	180	123	97		400

(a) 7-NN classifier (79.75% correct)

(b) Linear classifier (70.0% correct)

Table 4–11: Seascape: Confusion matrix for classifiers based on height and width features.

In fact, the main confusion, as suspected, was between the sailboat and the buoy classes <sup>7</sup>. This was first noted in Chapter 2 (see Table 2–2) and also in Chapter 3. The tables demonstrate that the better overall classification rates were achieved using features that discriminated better between sailboats and buoys, and that, for all the features, a nonlinear discrimination boundary was required.

Guess	Correct class			Total
	Sail	Motor	Buoy	
Sail	161	1	<b>5</b>	167
Motor	0	120	4	124
Buoy	<b>19</b>	2	88	109
Total	180	123	97	400

(a) 7-NN classifier (92.25% correct)

Guess	Correct class			Total
	Sail	Motor	Buoy	
Sail	137	0	<b>6</b>	143
Motor	1	116	3	120
Buoy	<b>42</b>	7	88	137
Total	180	123	97	400

(b) Linear classifier (85.25% correct)

**Table 4–12:** Seascape: Confusion matrix for classifiers based on 16 Gaussian features.

Guess	Correct class			Total
	Sail	Motor	Buoy	
Sail	170	1	4	175
Motor	0	119	0	119
Buoy	<b>10</b>	3	93	106
Total	180	123	97	400

(a) 7-NN classifier (95.5% correct)

Guess	Correct class			Total
	Sail	Motor	Buoy	
Sail	137	1	<b>6</b>	143
Motor	0	117	2	119
Buoy	<b>43</b>	<b>6</b>	89	138
Total	180	123	97	400

(b) Linear classifier (85.75% correct)

**Table 4–13:** Seascape: Confusion matrix for classifiers based on Wilks’-based Fourier features.

This situation, of differing separability complexity between classes in a multi-class ( $C > 2$ ) problem, is common in many real problems. For example, with the NIST data, as shown in Table 4–14 and Table 4–15, there were confusions between 0’s and 2’s, 7’s, and 8’s, as well as

<sup>7</sup>Off-diagonal confusions greater than 4 are marked in bold

between 9's and 4's, 7's and 8's. Using the zone features removed many of these problems but did actually increase the confusion between 4's and 9's.

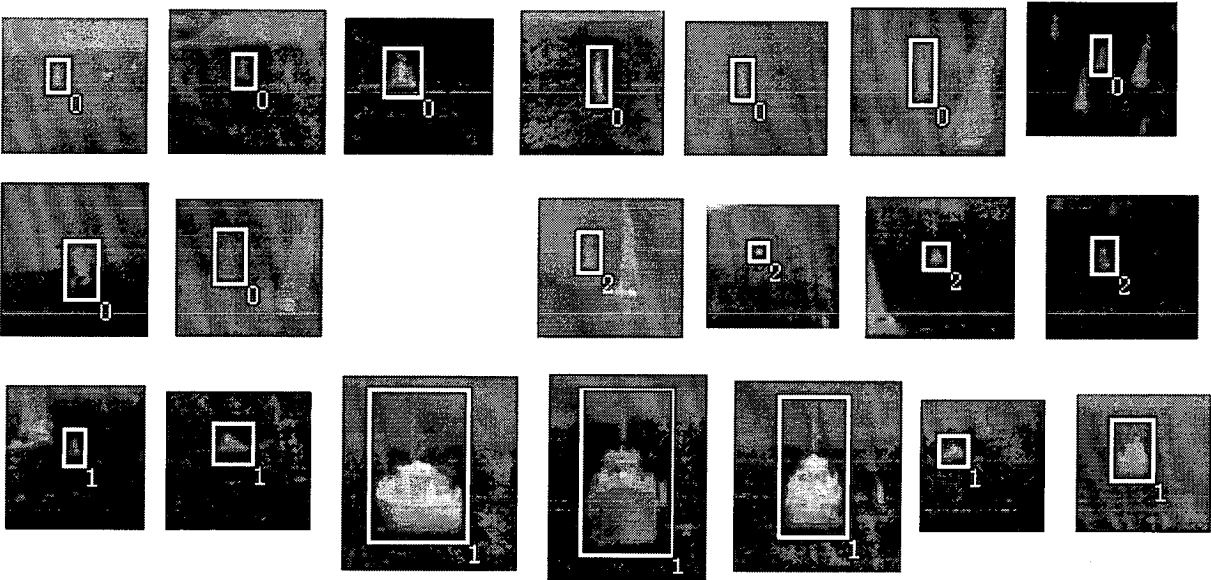
Guess	Correct class										Total
	0	1	2	3	4	5	6	7	8	9	
0	81	0	9	1	0	5	0	10	6	3	115
1	2	91	3	0	3	0	4	1	1	1	106
2	3	0	61	3	2	1	0	4	3	3	80
3	0	0	0	79	2	1	0	0	3	1	86
4	0	0	0	0	55	0	0	1	3	2	61
5	0	0	0	1	0	25	0	0	2	0	28
6	6	0	3	4	7	4	85	0	1	0	110
7	0	0	5	1	1	0	0	44	2	3	56
8	0	0	0	3	0	2	0	3	53	9	70
9	2	0	6	1	3	0	0	13	7	56	88
Total	94	91	87	93	73	38	89	76	81	78	800

**Table 4–14:** NIST: Confusion matrix for 7-NN classifier, 16 Gabor features (78.75% correct).

Guess	Correct class										Total
	0	1	2	3	4	5	6	7	8	9	
0	91	0	1	0	0	0	0	0	1	1	94
1	1	89	1	1	1	0	1	0	2	0	96
2	0	0	73	3	0	0	0	0	0	0	76
3	1	0	8	85	0	1	0	1	2	0	98
4	0	0	0	0	62	0	0	0	0	0	62
5	0	0	0	1	1	34	0	0	0	0	36
6	0	0	2	0	0	0	88	0	0	0	90
7	0	0	0	1	0	0	0	66	1	2	70
8	1	2	2	0	0	3	0	0	73	0	81
9	0	0	0	2	9	0	0	9	2	75	97
Total	94	91	87	93	73	38	89	76	81	78	800

**Table 4–15:** NIST: Confusion matrix for 7-NN classifier, 16 zone features (92.0% correct).

Returning to the seascape results it was interesting to examine exactly which buoys or sailboats were causing the confusion. It was thought that this would provide information on how to better separate them, or at least reason why, they were mistaken. Specifically, were there objects that were repeatedly misclassified, independent of classifier or feature used? In fact there were a base set of 20 objects that were repeatedly mistaken, and there were many others very similar which were frequently misclassified. Figure 4–19 shows the 20 objects discussed. Many of these objects are very small, or very thin, or have some other characteristic which makes discrimination exceedingly difficult, to the extent that even people find the task impossible. Also, note the ferry, in the bottom row, are the same objects from Figure 4–6.



**Figure 4–19:** Seascape: The rogues’ gallery - objects that were always misclassified.

### 4.6 Review

This chapter has examined the necessity and implementation of both feature extraction and classification. Various standard techniques were applied to both the real IR seascape data and the NIST digit database and the resulting features, to varying degrees, were successfully classified. For both the seascape, and NIST, databases a nonlinear solution provided a better solution in terms of the misclassification rate. For the seascape data, a 96% classification rate

was achieved. This outperformed the expert human classifier tested in Chapter 2 who scored 92%.

Furthermore, it was shown that applying feature selection techniques improved the choice of features to be used for a particular database. However, it was noted that these techniques are not infallible, they are not optimised with respect to a final classification error criterion and, often, they are compute and time intensive. A solution was to design a combined classification and feature extraction model that could be optimised in parallel, did not require *a priori* knowledge of the database or expert knowledge of feature extraction, yet maintained a controllable number of model parameters by making use of the correlated nature of the high dimensional pattern space. This model is the focus of the next chapter.

---

## Chapter 5

# Adaptive kernel neural networks

---

In the previous chapter various feature extraction methods were applied to both the character recognition and infrared seascape problems. The statistic chosen to select a suitable subset of features for classification was, as explained, quite naive. There were more complicated procedures available but a more attractive solution would be to automatically determine an adequate set of features in a combined feature extraction and classification model. Normally a neural network model, such as an MLP, would be an ideal solution. Unfortunately, the large dimensional input space of the image data usually prohibits this due to the subsequently large parameter vector required to be estimated in relation to a finite data set. Adaptive wavelet models in the last few years have been used to address this very problem [119,63,118,111,70]. This chapter reports on the application of adaptive wavelet technology on the both the seascape and NIST databases. This approach had never been used for classifying the type of real infrared data encountered in this project. Casasent *et al* used adaptive wavelets for the detection of real infrared objects [21,20], Szu *et al* used them for phoneme and speaker recognition [119], Shustorovich for character recognition [111], Kocur and Rogers for cancer diagnosis [63] and most recently Mallet *et al* applied them to mineralogical spectra data [70]. These authors used relatively complicated wavelets to adapt. However, in the project much simpler adaptive kernels were first tested before the wavelet models were used. The adaptive model was then extended, as suggested but not implemented by Szu *et al*, to incorporate a standard nonlinear layer to further improve generalisation [119]. The chapter begins with a look at kernel feature extraction.

## 5.1 Kernel feature extraction

Kernel feature extraction is the linear, sub-spatial transform of a correlated input space, such as an image, to a feature vector,  $\mathbf{d}$ , for the purpose of classification. Each element of the feature vector is generated using a *kernel*,  $\psi$ , which is characterised by its own set of parameters,  $\Phi = \{\phi_1, \phi_2, \dots, \phi_P\}$ , such that

$$d_i = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \psi(x, y; \Phi_i) dx dy \text{ for } i = 1, \dots, M \quad (5.1)$$

or in the discrete case, using a double summation approximation to the integrals, as <sup>1</sup>

$$d_i = \sum_{x=1} \sum_{y=1} f(x, y) \psi(x, y; \Phi_i). \quad (5.2)$$

The kernel parameters control shape, position, and scale and the larger the value of  $P$ , the greater the flexibility of each kernel. These parameters subsequently control the classification potential of the generated features.

Kernel feature extraction is very common and many of the techniques discussed in Chapter 4 can be expressed in terms of kernel feature extraction. This is demonstrated in Table 5–1.

Of course, there are many other types of feature extraction algorithms but many of these procedures can not, like kernel feature extraction, be so easily combined with a standard MLP model. Due to the linearity of the kernel transform, and the linear first layer of the MLP model, the kernel feature extraction and MLP model can be expressed as a single entity relating image input directly to the required classification output. This is described in Equation 5.3

$$z_k(f; \Phi') = w_{0k} + \sum_{j=1}^N w_{jk} \varphi \left( w_{0j} + \sum_{i=1}^M w_{ij} \sum_x \sum_y f(x, y) \psi(x, y; \Phi_i) \right) \quad (5.3)$$

where  $N$  represents the number of hidden units with the  $\varphi$  nonlinearity, and  $\Phi'$  represents the full classification model parameter vector and is comprised of the weights and biases,  $w_{jk}$ , and the  $P$ -dimensional kernel parameter vectors,  $\Phi_j$ .

---

<sup>1</sup>The Cartesian coordinates  $x, y$  will be used in conjunction with the  $x, y$  digital indices with hopefully little confusion.



Feature extraction method	Kernel, $\psi(x, y)$	Notes
Zoning	1	Defined over the region: $x_1 \leq x < x_2$ $y_1 \leq y < y_2$
Projection histogram (in $x$ or $y$ )	1	Defined over the region: $x < x_2 \forall y$ or $y < y_2 \forall x$
Geometric moments	$x^p y^q$	Moment of order, $p + q$ .
Fourier	$e^{-i(ux+vy)}$	A complex kernel. Features used include the magnitude or $\Re/\Im$ complex pairings.
Cosine transform	$\alpha(u)\alpha(v)\cos[(2x + 1)u].$ $\cos[(2y + 1)v]$	The parameters $u$ and $v$ , as with the Fourier transform, control spatial frequency and orientation of the kernel.

Table 5–1: Examples of feature extraction kernels

This ability to combine the feature extraction and classification, that were treated as separate processes in Chapter 4, is very important as it implies that the combined model may be optimised directly against the object image data. This direct optimisation means that the features should be optimal, with respect to the output classification error criterion, for each new database.

## 5.2 Adaptive kernel feature extraction

In Chapter 4 the zoning technique was applied to images using a set of non-overlapping, uniform, square, identical kernels that covered the entire input space. Each kernel had a discontinuity at the boundary and this made the features highly sensitive to small distortions, or shifts, by the object around these areas. This would be especially noticeable with binary images. Furthermore, the uniformity of the kernel assumed equal importance to all image pixels within the region covered by the kernel. The square shape of this region was also arbitrary and the necessity to cover the entire image was inefficient. To solve this latter issue Chapter 4 demonstrated a simple, but laborious, method of determining a subset of features best suited for a specific problem. However, a better solution was to have a fixed number of kernels, or even better, a linear superposition of kernels, that could in some way adapt their positions and shape according to an overall classification error criterion for each specific problem. This had already been addressed with adaptive wavelet theory [120].

### 5.2.1 Adaptive wavelets

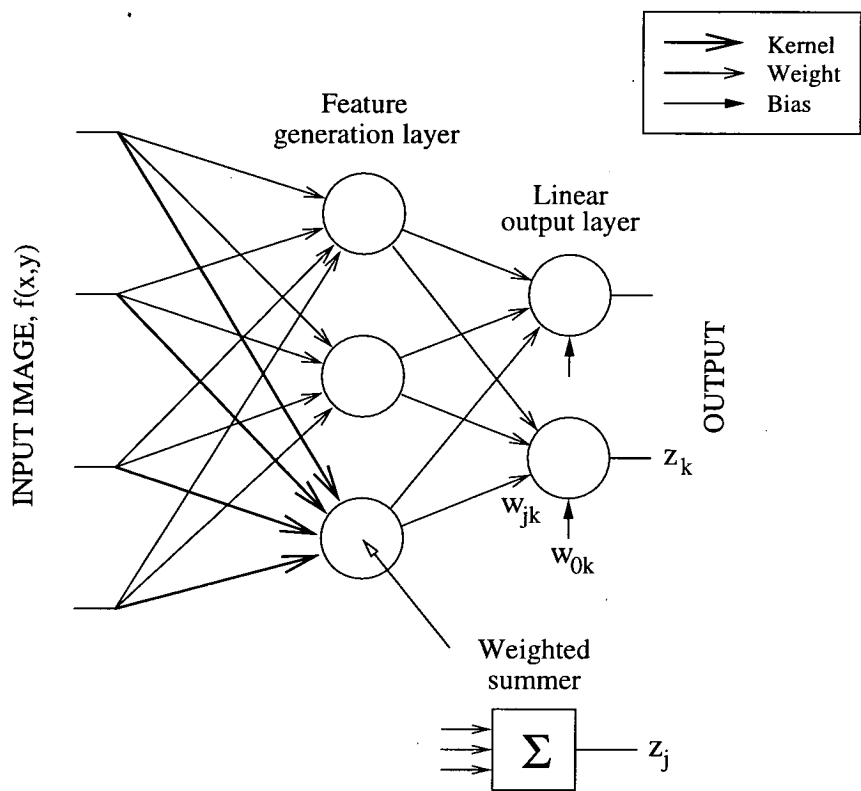
The "super-wavelet" concept was introduced by *Szu et al.* as a combination of adaptive wavelet feature extraction and linear class discrimination and was applied successfully to problems of signal representation and classification [119]. Many of the problems of feature selection were circumvented by this concept of a "super-wavelet" due to the direct adaptation of the feature extraction, whilst maintaining a controllable numbers of adjustable parameters.

The "super-wavelet" is a linear weighted sum of  $M$  adaptive wavelets which are shifted and dilated versions of a mother wavelet,  $\psi$ . To classify a two-dimensional signal, such as an image  $f(x, y)$ , a linear discriminant of the form

$$z_k(f; \Phi') = w_{0k} + \sum_{j=1}^M w_{jk} \sum_x \sum_y f(x, y) \psi(x, y; \Phi_j) \quad (5.4)$$

can be implemented where  $z_k$  represents one of  $C$  classifier outputs and the full classification parameter vector,  $\Phi'$ , is comprised of the weights and biases,  $w_{jk}$ , and the  $P$ -dimensional

kernel parameter vectors,  $\phi_j$ . Hence, in the model there are  $T = PM + C(M + 1)$  adaptive parameters. This is simply a linear version of Equation 5.3 and as such the adaptive wavelets can simply be considered as a subset of the adaptive kernel models proposed in this thesis. Figure 5–1 shows a diagrammatic representation of Equation 5.4.



**Figure 5–1:** Architectural representation of a linear adaptive wavelet (kernel) classifier with one kernel,  $\psi$ , highlighted in bold. Input images are multiplied by a kernel and summed to generate features in the first layer. The second layer acts as a simple linear discriminant.

5.2.2 Error derivatives

To estimate the parameter vector,  $\hat{\Phi}'$ , that minimised a classification error criterion, such as sum-of-squares, using traditional line-searching techniques required calculation of the model error derivatives.

The estimate of the parameter vector,  $\hat{\Phi}'$  for the models given in Equation 5.4 and Equation 5.3, were derived by optimisation with respect to an output classification error criterion,  $E$ .

A conjugate gradient directed line searching technique was used to determine the estimate (see Appendix A), as it was used in Chapter 4 to estimate traditional MLP classifier parameters. As with the MLP, the conjugate gradient method required knowledge of the error derivatives, for example  $\partial E / \partial \phi_{jp}$ , for the linear network of Equation 5.4. This section shows how both the first order error derivatives and the second order, Hessian, matrix of error derivatives for the linear network were derived. This was easily extended to the nonlinear model, using standard backpropagation procedures which can be found in Bishop, page 140 [13].

The linear model was first simplified to

$$z_k = w_{0k} + \sum_{j=1}^M w_{jk} z_j \quad \text{where} \quad z_j = \sum_x \sum_y f(x, y) \psi(x, y; \Phi_j)$$

and the error for each pattern in the training set was given as  $E_n$  such that  $E = \sum_n E_n$ . Immediately the output layer error derivatives could be described as

$$\frac{\partial E_n}{\partial w_{jk}} = \frac{\partial E_n}{\partial z_k} \cdot z_j$$

and the output bias could simply be treated as a weight but with  $z_{j=0} = 1$ . The kernel parameter error derivatives were given by

$$\frac{\partial E_n}{\partial \phi_{jp}} = \sum_{j'} \frac{\partial z_{j'}}{\partial \phi_{jp}} \cdot \frac{\partial E_n}{\partial z_{j'}} = \frac{\partial z_j}{\partial \phi_{jp}} \cdot \frac{\partial E_n}{\partial z_j}$$

and by expanding  $\partial E_n / \partial z_j$

$$\frac{\partial E_n}{\partial \phi_{jp}} = \frac{\partial z_j}{\partial \phi_{jp}} \sum_k \frac{\partial E_n}{\partial z_k} \cdot w_{jk}$$

These were the simple first order derivatives which could be used in the gradient based minimisation algorithms. The Hessian matrix,  $\mathbf{H}$ , was useful for determining the *conditioning* of the optimisation. The condition number of a Hessian is the ratio of the largest Hessian eigenvalue,  $\lambda_{max}$ , to the smallest,  $\lambda_{min}$ . A large number would indicate ill-conditioning in the optimisation process and consequently large numbers of training iterations.

For the output layer weights

$$\frac{\partial^2 E_n}{\partial w_{j'k'} \partial w_{jk}} = \frac{\partial}{\partial w_{j'k'}} \left( \frac{\partial E_n}{\partial z_k} \cdot \frac{\partial z_k}{\partial w_{jk}} \right) =$$

$$\frac{\partial z_k}{\partial w_{jk}} \cdot \frac{\partial^2 E_n}{\partial w_{j'k'} \partial z_k} = \frac{\partial z_k}{\partial w_{jk}} \cdot \frac{\partial z_{k'}}{\partial w_{j'k'}} \cdot \frac{\partial^2 E_n}{\partial z_{k'} \partial z_k}$$

This was rewritten as

$$\frac{\partial^2 E_n}{\partial w_{j'k'} \partial w_{jk}} = z_j z_{j'} \delta_{kk'} \cdot \frac{\partial^2 E_n}{\partial z_k^2}$$

where  $\delta_{kk'}$  is the Kronecker delta symbol.

A similar process was applied to the kernel parameters and also the combination of both weight and kernel parameters, such that

$$\frac{\partial^2 E_n}{\partial \phi_{j'p'} \partial \phi_{jp}} = \frac{\partial^2 z_j}{\partial \phi_{j'p'} \partial \phi_{jp}} \sum_k w_{jk} \cdot \frac{\partial E_n}{\partial z_k} + \frac{\partial z_j}{\partial \phi_{jp}} \cdot \frac{\partial z_{j'}}{\partial \phi_{j'p'}} \cdot \sum_k w_{jk} w_{j'k} \cdot \frac{\partial^2 E_n}{\partial z_k^2}$$

and also

$$\frac{\partial^2 E_n}{\partial \phi_{j'p'} \partial w_{jk}} = \frac{\partial^2 E_n}{\partial w_{jk} \partial \phi_{j'p'}} = \frac{\partial z_{j'}}{\partial \phi_{j'p'}} \left\{ z_j w_{j'k} \cdot \frac{\partial^2 E_n}{\partial z_k^2} + \frac{\partial E_n}{\partial z_k} \cdot \delta_{jj'} \right\}$$

The Hessian matrix was thus determined as

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 E_n}{\partial w_{j'k'} \partial w_{jk}} & \frac{\partial^2 E_n}{\partial \phi_{j'p'} \partial w_{jk}} \\ \frac{\partial^2 E_n}{\partial w_{j'k'} \partial \phi_{jp}} & \frac{\partial^2 E_n}{\partial \phi_{j'p'} \partial \phi_{jp}} \end{bmatrix}. \quad (5.5)$$

In the the initial experiments on the seascape and NIST data a simple sum-of-squares error criterion,  $E^{SSE}$ , was used where  $E_n^{SSE} = 0.5 \sum_k (t_k - z_k)^2$  and  $t_k$  is the target value for the  $n$ th pattern. This simplified calculation of the Hessian as for the SSE  $\partial^2 E_n / \partial z_k^2 = 1$ . The Hessian was derived as it could be potentially used for many purposes including second order nonlinear optimisation, identifying least significant parameters in a classification model, and for determining regularisation parameters. For further details see Bishop, page 150 [13].

Using these error derivative calculations kernels could be estimated that minimised  $E^{SSE}$ . But, as stated by Daugman [26] these resulting feature extractors,  $\hat{\psi}$ , were required to be neither orthogonal ( $\langle \hat{\psi}(x, y; \Phi_j), \hat{\psi}(x, y; \Phi_k) \rangle \neq 0$  for all  $j \neq k$ ) nor complete in order to satisfy optimality according to  $E$  and the main consideration was the form of  $\psi$ .

5.2.3 Constraints on the form of  $\psi$

Some examples of potential adaptive kernels,  $\psi$ , have been introduced, such as the adaptive wavelets. However, no constraints have yet been placed on the form of the kernel. The following restrictions were placed on the kernels to be used in the project. These will be valid for other projects.

- Flexibility over the image space parameterised by a finite parameter set,  $\Phi$ .
- There must be no element of  $\phi \in \Phi$  such that  $\psi' = \phi\psi$ .
- $\psi$  must be differentiable with respect to  $\Phi$
- $\int \psi < \infty$  over the image space.

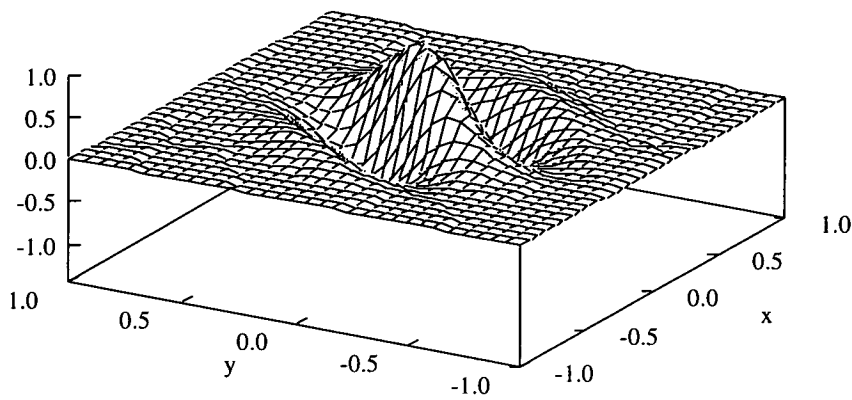
5.2.4 Kernel selection

Many authors have used the real, imaginary, or complex Gabor transform as a suitable kernel and have successfully applied it to many problems including image representation [26], object detection [21,20] and character recognition [111]. The Gabor transform is given by

$$\psi(x, y; x_0, y_0, a, b, u, v) = \exp\{ -[(x - x_0)^2 a^2 + (y - y_0)^2 b^2] \} \cdot \exp\{ -2\pi i [u(x - x_0) + v(y - y_0)] \} \tag{5.6}$$

and is comprised of a Gaussian, centred at  $(x_0, y_0)$  and with scaling values  $(a, b)$ , modulated with a complex exponential with spatial frequency  $(u^2 + v^2)^{1/2}$  and orientation  $\arctan(v/u)$ . An example of the real part of a typical kernel is given in Figure 5–2 as this is often used in object recognition whilst the imaginary part used for segmentation or boundary detection.

For this project other, simpler, kernels were implemented, as well as the Gabor kernel. These kernels appeared not have been tested in the literature. The reason for examining the simpler kernels was that even though the Gabor transform has many attractive properties it can be approximated by a linear summation of these much simpler kernels. These kernels can also



**Figure 5–2:** Example of the real part of Gabor transform.

approximate a wide range of other functions, as well as Gabor, by simple parameter adjustment. This approach of using many simple kernels instead of a few complex kernels has been widely used in kernel-based density estimation [103].

The problem of selecting the relevant features for a specific problem switched to one of choosing from a set all possible kernels,  $\Psi$ , the type of kernel,  $\psi \in \Psi$ , the number of kernels, and which kernel parameters to adapt in the model. Furthermore, the form of each individual kernel may influence classification, i.e. using  $\psi_j(x, y; \Phi_j)$  instead of  $\psi(x, y; \Phi_j)$ . With the exception of the last issue due to time limitations, these issues were investigated, and applied to, the NIST and seascape databases.

## 5.3 Adaptive kernel experiments

In the following sections various adaptive kernel neural networks, of the type given in Equation 5.4 and Equation 5.3, were applied to both the NIST and seascape databases. Various different kernels, and kernel parameters, were tested, and problems discussed.

### 5.3.1 Linear classification

The first section concentrates on the combined feature extraction with a linear discriminant model. This type of classifier, as stated previously, has been popular with many researchers and results have been published for adaptive wavelet kernels. The first kernel to be tested was a simple Gaussian of variable width.

#### Fixed position kernel adaptation

In Chapter 4 sixteen fixed position Gaussian kernels, with a suitably fixed width,  $a$ , were used as kernels. Results for the seascape data were impressive with high classification rates achieved: 86.0% using a linear classifier. The single value of width chosen though was quite arbitrary. A better idea was for each kernel to have an individual  $a$ , reflecting property changes across the image, and for those widths to be determined automatically. This was available with the adaptive networks that have been described in this chapter.

Experiments were performed using different numbers of Gaussian kernels, fixed in a regular square format across the image space, with either one or two width parameters per kernel, the latter controlling width in both the  $x$  and  $y$  direction. Widths were initialised randomly per kernel using a uniform distribution, of unit variance, about a mean value of 2.5. The results, averaged over 10 experiments, for different numbers of kernels are given in Tables 5–2



Number of kernels, $M$	Fixed features	$P = 1$ ( $a = b$ )	$T$	$P = 2$ ( $a, b$ )	$T$
4	67.25 (1.4)	68.5 (1.0)	19	69.0 (1.4)	23
9	84.5 (1.8)	87.75 (1.1)	39	88.75 (1.2)	48
16	86.0 (1.5)	88.5 (1.5)	67	<b>90.0</b> (1.7)	83
25	85.5 (1.2)	88.5 (1.8)	103	89.5 (1.3)	128

**Table 5–2:** Seascape: Adapting Gaussian variance parameters. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests. The value  $T$  represents the total number of parameters in the model.

and 5–3. The total number of parameters,  $T$ , in each model are also given. The usual MLP training conditions were applied such as data splitting, and early stopping.

Number of kernels, $M$	Fixed features	$P = 1$ ( $a = b$ )	$T$	$P = 2$ ( $a, b$ )	$T$
4	49.0 (1.0)	63.25 (1.4)	54	61.0 (3.7)	58
9	68.25 (1.3)	74.5 (2.4)	109	75.75 (2.8)	118
16	76.25 (1.5)	83.5 (1.8)	186	79.0 (2.0)	202
25	83.25 (1.2)	<b>87.5</b> (1.1)	285	85.25 (4.8)	310

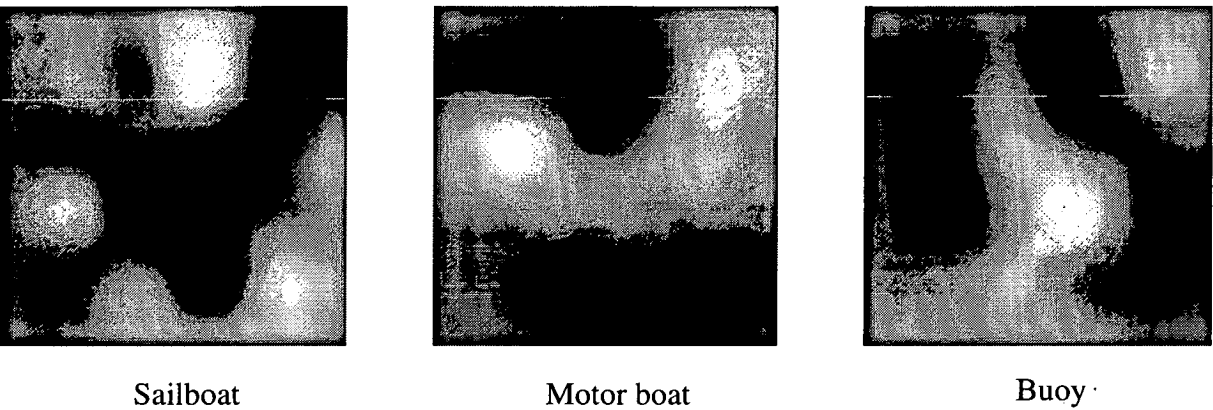
**Table 5–3:** NIST: Adapting Gaussian variance parameters. Each score is the mean percentage classification over 10 different samples each consisting of 800 test vectors. The value in parentheses is the standard deviation over the 10 tests. The value  $T$  represents the total number of parameters in the model.

With both the seascape and NIST databases improvements in classification using the adaptive variance kernels were recorded. Naturally, as the number of kernels increased the difference between the fixed and the adaptive kernels was less marked. Adding the second variance parameter had little, even some detrimental, effect when using fixed position kernels. This suggested

that the extra kernel parameter was increasing the dimensionality of the error surface yet was unable to reduce the global minimum of the surface. Bidirectional adaptability of the fixed position kernels did little to aid classification for both databases.

The condition number of the error Hessian is a measure of how ill-conditioned is the model, and is determined by calculating the absolute value of the ratio of the maximum to minimum eigenvalue of the error Hessian,  $H$ . It is well known that MLP's are poorly conditioned [93], and a quick test was required to check that these new adaptive models were no worse. A condition number of approximately  $e^7$  is not uncommon with MLP's and this appeared to be the same for the adaptive networks.

Figure 5–3 and Figure 5–4 display, as a set of images, the resulting *super-kernels* (the weighted sum of all kernels) for each class in both the seascape and NIST databases. One



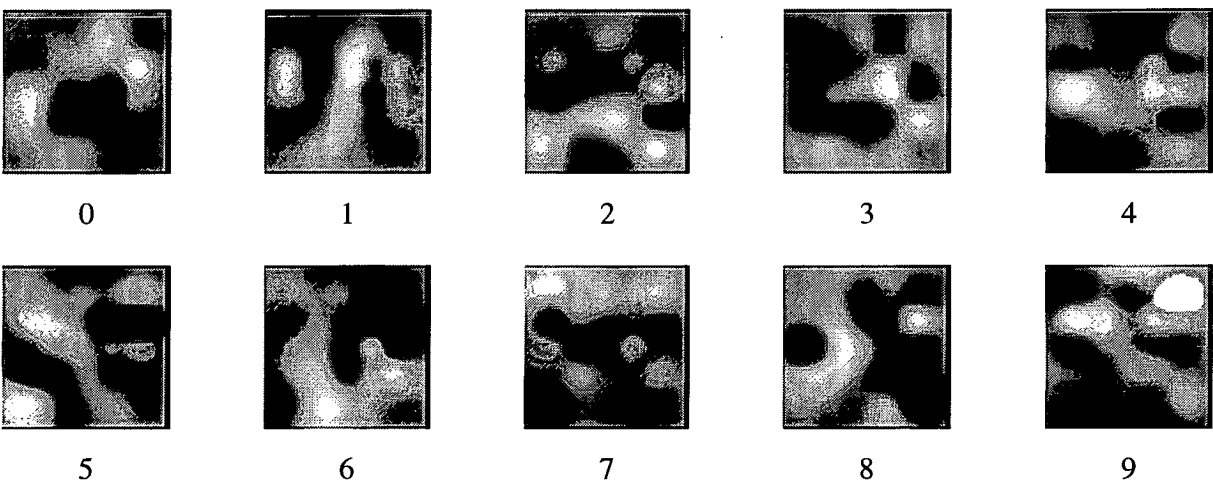
**Figure 5–3:** Seascape:  $M = 16$  adaptive Gaussian variance resulting super-kernels. Bright areas represent image locations where the effect on the final classification of an object is biased towards the class of the super-kernel (positive effect), grey areas are where objects do not effect the class decision (nil effect) and black where the effect an object, at that location, is against the super-kernel class decision (negative effect).

variance per kernel was used. The regular kernel position pattern can be easily seen with bright spots representing large kernel magnitude. The classification outputs are derived by correlating each input image with each super-kernel. The largest correlation is the predicted class. Thus from examining the super-kernels it was possible to determine how each type of object was being classified.

With the seascape data sailboats were identified by the tops of their masts (in either vertical or slanting mode), and the hull. The motor boats were identified simply by their thin horizontal nature, especially at the left and right extremities. The buoys were recognised using central image data.

The NIST results showed that some of the resulting super-kernels were working like a standard correlator, especially for the digits zero to three. The others digits were more complicated. The digit eight was only represented in the left half of the super-kernel. If a whole eight were used, as in a standard correlator, significant response would have occurred by a number three class object. By using only the left half region, the 3-8 confusion was significantly reduced. This feature was automatically generated by the adaptive model.

Sevens were predominantly characterised by a strong horizontal line at the top of the image, whilst sixes and nines where identified by strong energy responses in the bottom left, and top right regions respectively.



**Figure 5–4:** NIST:  $M = 25$  adaptive Gaussian variance resulting super-kernels. Note the images likeness, or partial likeness, to individual digits.

Adaptive kernel positioning

The previous section demonstrated that the adaptive model did work. The kernel parameters chosen though were not particularly effective. A more productive approach was to allow the kernels flexibility to move, and concentrate on regions of image space rich in discriminatorial information. This was done by adapting the kernel centres,  $(x_0, y_0)$ .

One of the first problems encountered with this approach was that the model would not appear to optimise, only adapting such that one class was always predicted. The problem was the kernel centre derivatives were dominant initially, and with random output weights, the kernels moved out of the image producing a null feature vector,  $d = 0$ . This was solved by allowing the output weights to adapt on their own first for a few iterations before allowing combined kernel and output weight adaptation.

This proved successful, and the results are shown in Tables 5–4 and 5–5 for both seascape and NIST databases.

Number of kernels, $M$	Classification %	T
3	86.25 (1.3)	18
6	89.5 (1.1)	33
9	90.75 (1.0)	48
12	91.25 (1.2)	63
15	91.25 (1.1)	78

**Table 5–4:** Seascape: Adapting Gaussian kernel positions. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests. The value T represents the total number of parameters in the model.

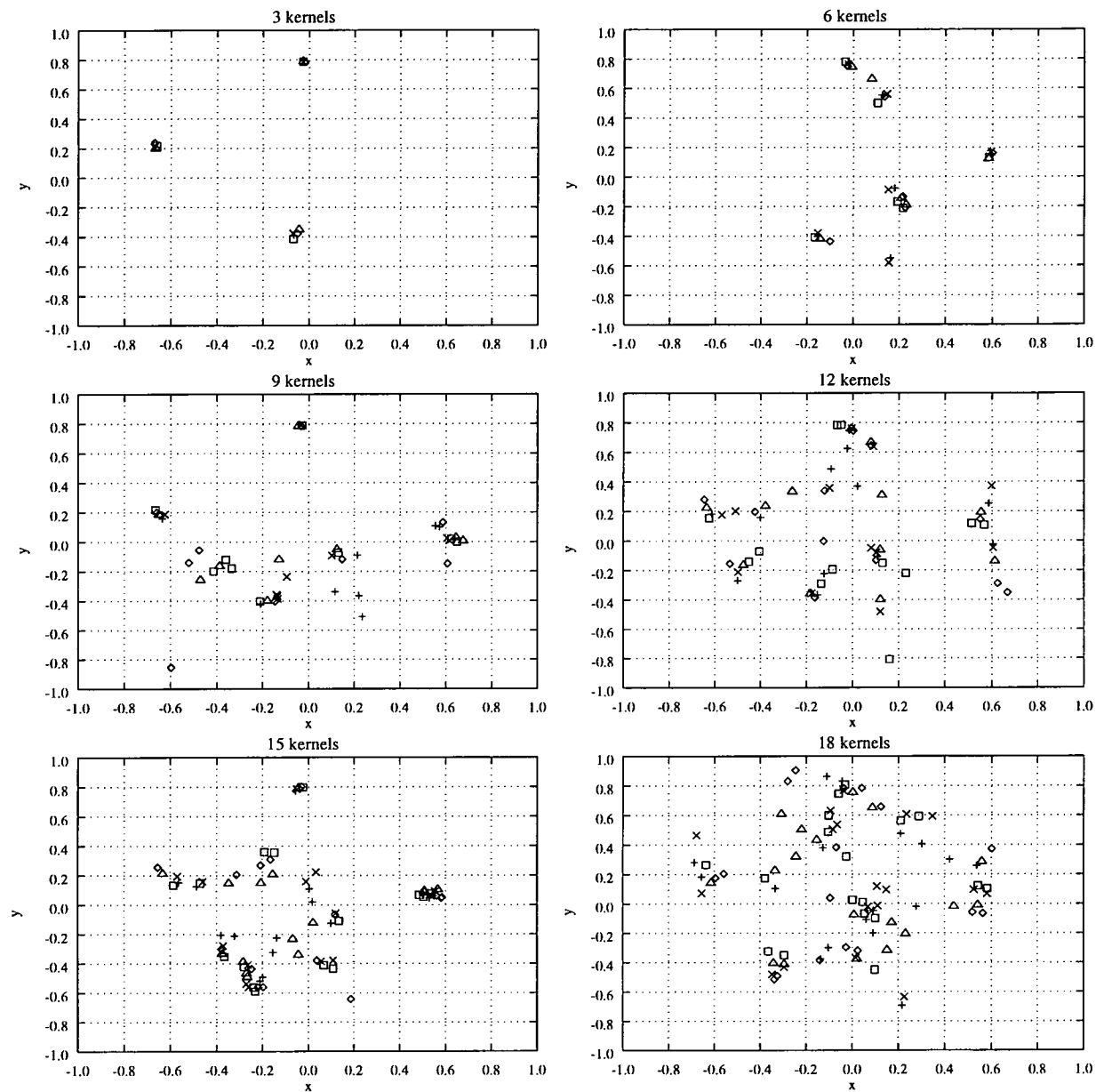
Number of kernels, $M$	Classification %	T
3	54.0 (4.3)	46
6	71.75 (2.4)	82
9	78.0 (2.1)	118
12	80.75 (1.8)	154
15	83.75 (2.0)	190
18	86.0 (2.6)	226
21	86.75 (1.9)	262
24	87.0 (2.4)	298

**Table 5–5:** NIST: Adapting Gaussian kernel positions. Each score is the mean percentage classification over 10 different samples each consisting of 800 test vectors. The value in parentheses is the standard deviation over the 10 tests. The value T represents the total number of parameters in the model.

The seascape results were very promising. Even with only three 2-parameter kernels the classification rate equals the 25 kernel solution in the previous section. Adding more kernels with this database though appeared to have little effect, with an increase of less than 1% between 6 and 15 kernels. However, with the NIST database, with many more classes to separate, the number of kernels required increased. In fact, the 25 fixed centre, single parameter, kernel performed better than any of the adaptive position models for the NIST data. It was thought that the size of the adaptive centres were too large, and combined with the large number of kernels required to suitably solve the problem, allowed for little flexibility of movement.

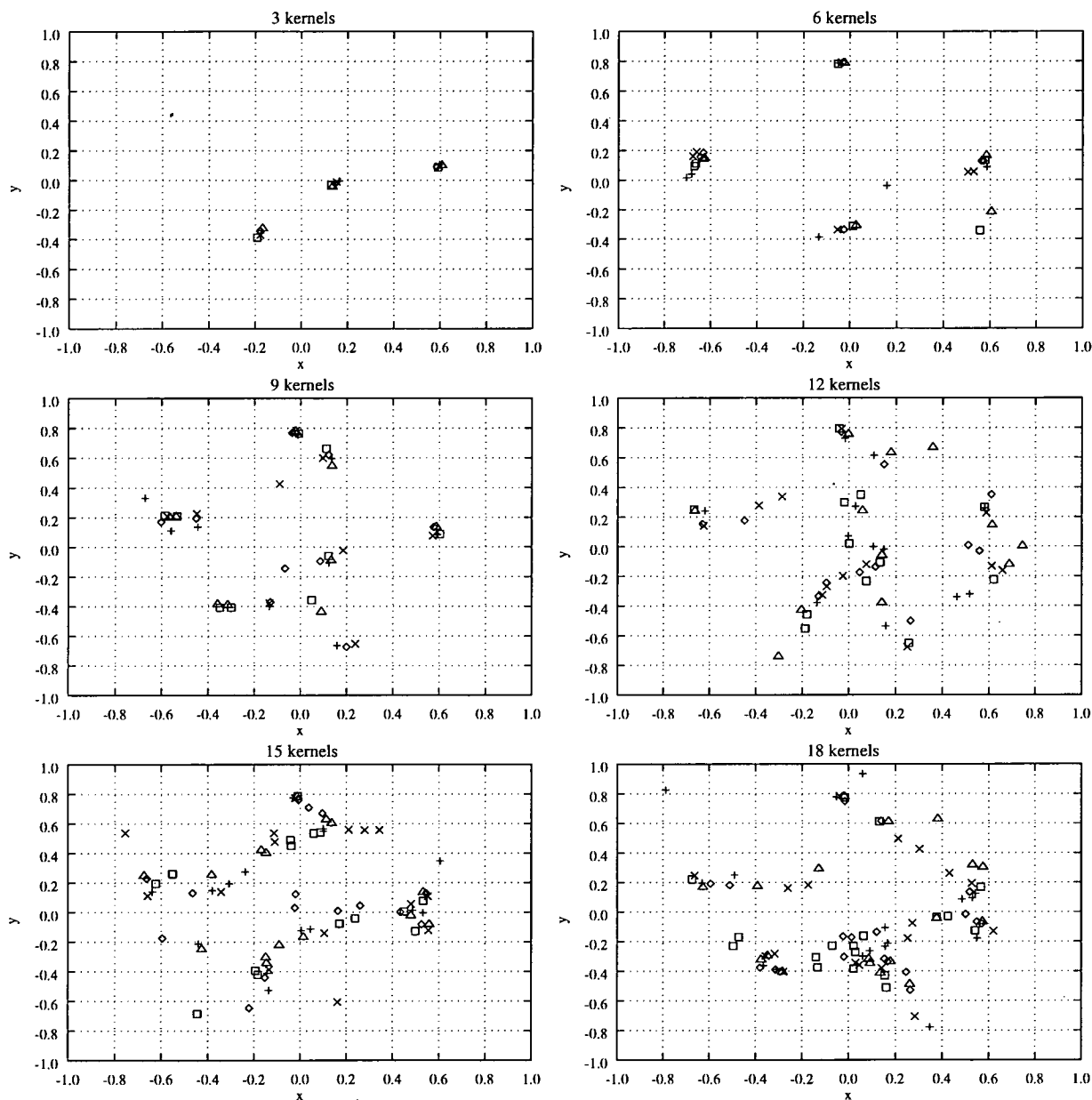
Once these experiments were completed an analysis of the final resting positions of the kernels was performed. Figure 5–5 shows the final positions of the kernel centroids using different number of kernels, and different splits of the seascape database. Figure 5–6 displays the results of a similar set of experiments, but where different starting positions were chosen.

The results from these experiments were interesting. For 3 kernels Figure 5–5 shows three distinct positions generated by different splits of the object database. Using identical data but different starting positions for the three kernels, as in Figure 5–6, results again



**Figure 5–5:** Seascape: Final centre positions of  $(x_0, y_0)$  parameter vector from the adaptive kernel positioning experiment. Identical kernel starting positions but different splits of the object database. Key:  $\triangle, \square, \diamond, +, \times$  represent the results from different splits of the data.

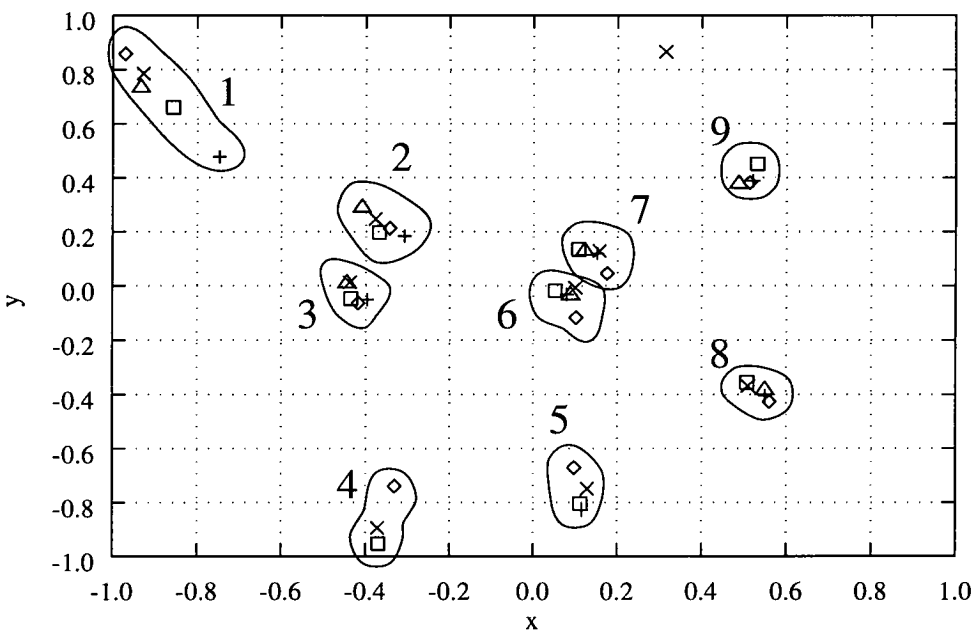
in three distinct positions, but two of the three being at different locations. It was found that when repeating with more starting positions the 3 kernels always finished in one of 5 locations. However, as the number of kernels increased the distinctiveness of these final



**Figure 5-6:** Seascape: As with the previous Figure but a different starting position. Again, different splits of the object database were used. Key:  $\triangle$ ,  $\square$ ,  $\diamond$ ,  $+$ ,  $\times$  represent the results from different splits of the data.

positions rapidly decreased. Different splits of the data found very different final resting locations. The corresponding classification results in Table 5-4 show that after 6 kernels

no appreciative increase in classification was achieved. It is suggested that there is a strong similarity here between under- and over-fitting in nonlinear models with too few kernels (lack of flexibility) relating to under-fitting and too many kernels (not generalising and fitting to individual data splits) relating to over-fitting. It might be expected then, after reviewing the NIST results, that with the NIST database that there would be greater distinctiveness with larger numbers of kernels. This was indeed true, and the final positions with the 9 kernel NIST model are given in Figure 5–7.



**Figure 5–7:** NIST: Final centroid positions for 9 kernel model. Key:  $\triangle$ ,  $\square$ ,  $\diamond$ ,  $+$ ,  $\times$  represent the results from different splits of the data.

Returning to the 6 kernel model in Figure 5–5 it appeared that the 5 positions discovered by the 3 kernel experiments were not all covered by the 6 available kernels. In fact, it appeared that there were two kernels residing at the same location. This produced very similar features and subsequently redundancy. From examining the trajectory plots it was found that improper initialisations of the kernel centres often were to blame. Kernels in close proximity were not diverging during optimisation, although extensive, further optimisation showed the kernels eventually diverging. This suggested that the error derivatives for the kernel positions were similar, and due to the shape of the error surface, were having difficulty separating.



Regularised kernel positioning

A solution to the divergence problem, given above, was to choose sensible starting positions for the kernels. This worked well for small numbers of kernels but as they increased in multitude the probability of trajectories converging appeared to increase. This is shown in Figure 5–8 where 6 of a total of 15 kernels are shown. The kernels have diverged, and in one case appears to have been deflected back into the path of another kernel from which it had been previously diverging.

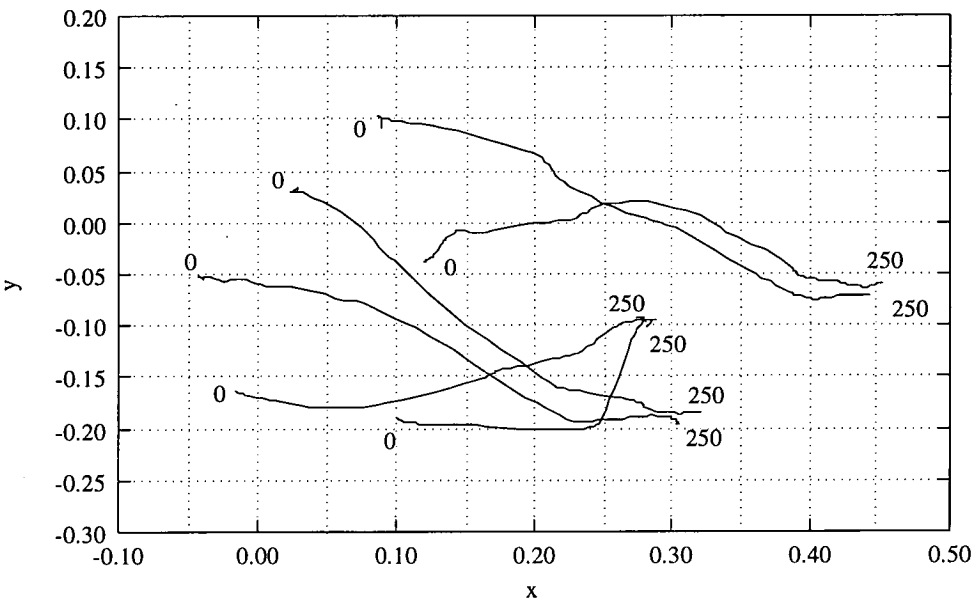


Figure 5–8: Seascape: Kernel centre trajectories problem.

Another solution was attempted which combined sensible kernel initialisation with a regularising penalty term in the error function, such as

$$E' = E + \frac{2\lambda}{M(M-1)} \sum_{j=1}^M \sum_{k=j+1}^M \Omega(x_{0j}, y_{0j}, x_{0k}, y_{0k}) \tag{5.7}$$

where

$$\Omega(x_{0j}, y_{0j}, x_{0k}, y_{0k}) = \exp(-((x_{0j} - x_{0k})^2 + (y_{0j} - y_{0k})^2)/\sigma_p^2)$$

and  $\sigma_p$  is the penalty variance which determines the proximity in which other kernels may reside without the error being penalised significantly. The  $\lambda$  term controlled how much importance

was placed on the penalty term. Examining the error derivatives showed that for each particular centre parameter, the penalty added a term which was a weighted sum of the distances between that particular centre and every other, in either the  $x$  or  $y$  directions. The weights being proportional to  $\Omega$ . For example,

$$\frac{\partial E'}{\partial x_{0n}} = \frac{\partial E}{\partial x_n} + \frac{4\lambda}{M(M-1)\sigma_p^2} \sum_{j=1}^M \Omega(x_{0j}, y_{0j}, x_{0n}, y_{0n})(x_{0j} - x_{0n}) \tag{5.8}$$

In practice, the two parameters,  $\lambda$  and  $\sigma_p$ , were combined into a single  $\lambda' = \lambda e^{1/\sigma_p^2}$  parameter which was set as to prevent convergence, but not discourage extreme kernel divergence. Figure 5-9 shows the penalty term being put to effect.

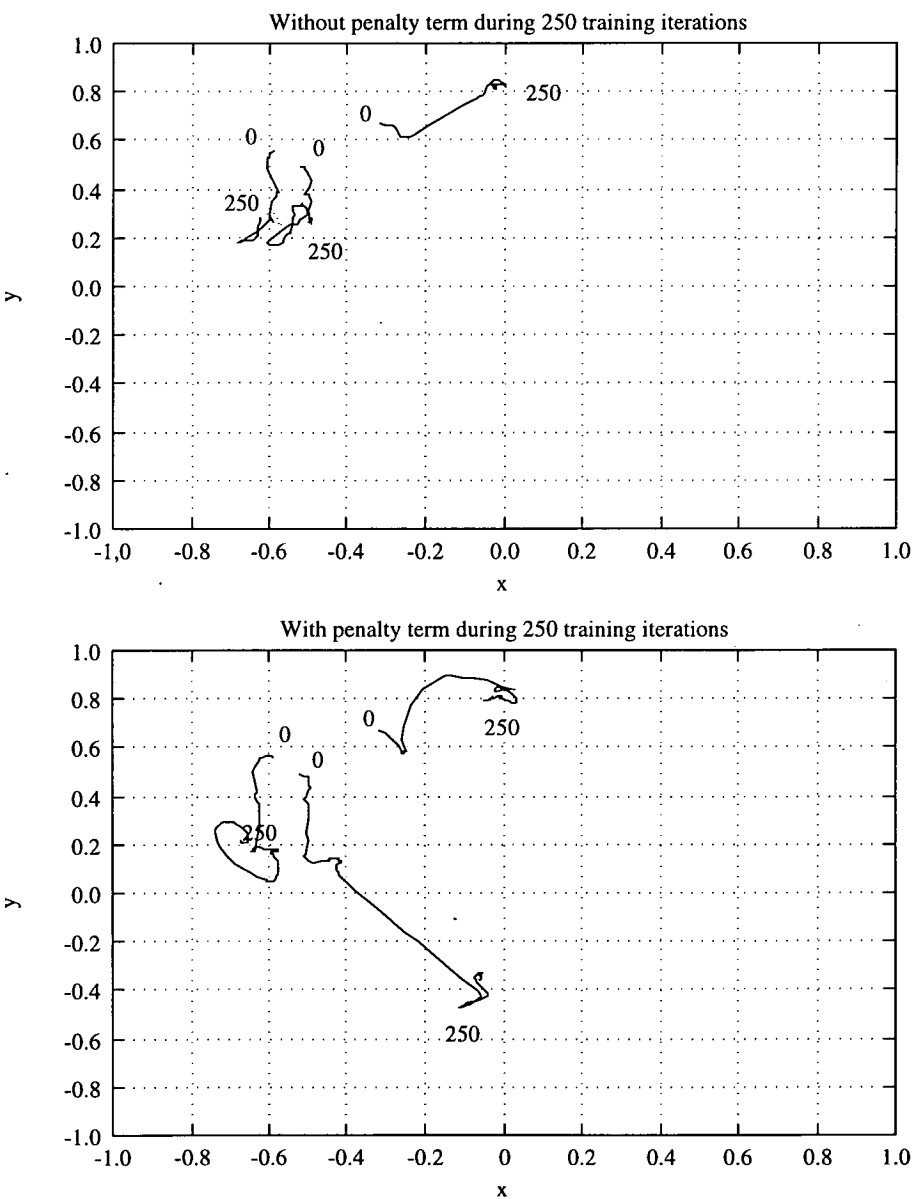
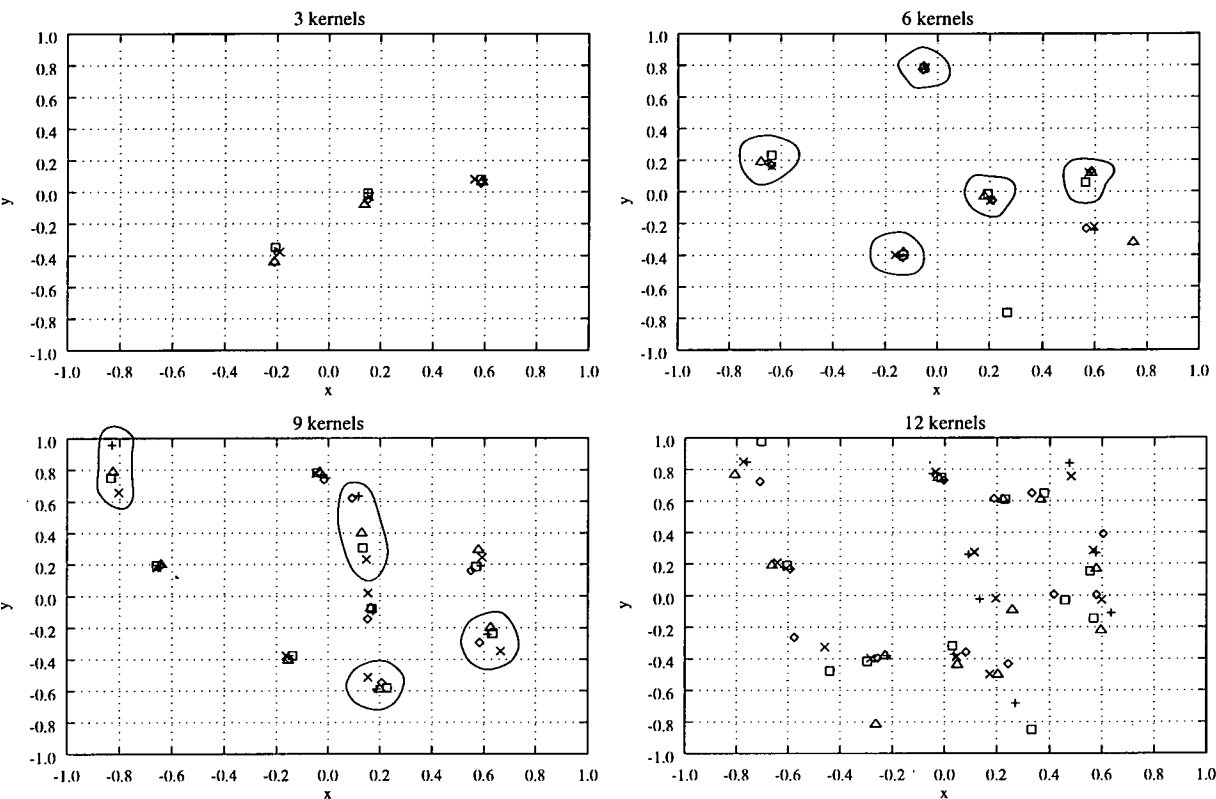


Figure 5-9: Examples with and without penalty influence.

This was an acceptable practical solution, although there was now another parameter controlling the performance of the classifier. The final centre positions using the penalty are shown in Figure 5–10. Note the 5 positions are now shown, and even the 9 kernel results with apparently 4 extra distinct final resting positions. Both the 6 and 9 kernel models with the added penalty term gave 1.5% increases in performance. The other models remained unchanged.



**Figure 5–10:** Seascape: Final centre positions of  $(x_0, y_0, \lambda')$  parameter vector using regularised kernel positioning. The same initial starting conditions were used but with different splits of the object database. Key:  $\triangle, \square, \diamond, +, \times$  represent the results from different splits of the data.

Another method for preventing feature collinearity was to ensure that each kernel had a unique shape, such that, even if they were locally identical, the features produced would be dissimilar. Thus, the next type of kernel tested included four parameters, the two Gaussian widths, as well as the two centres, for each kernel.

**Adapting shape and location**

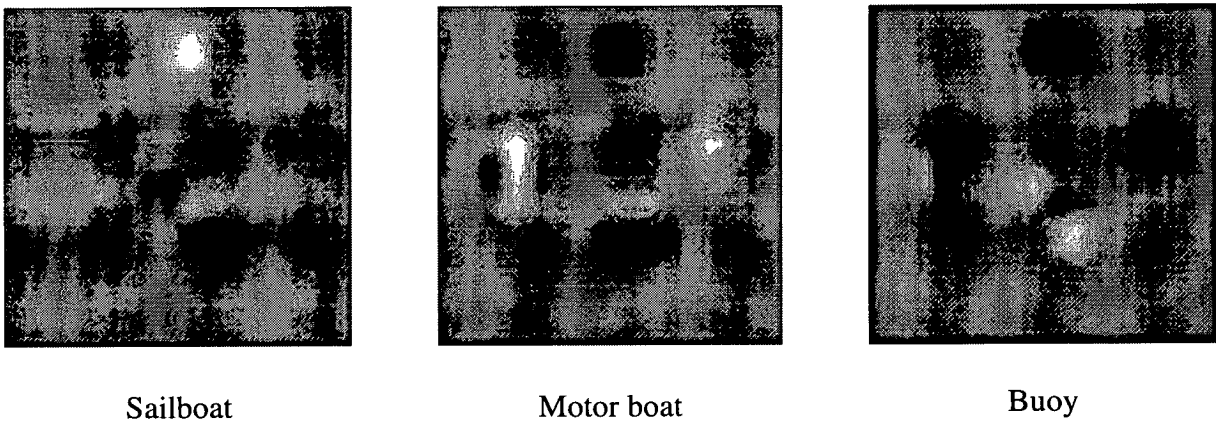
Tables 5–6 and 5–7 provide the classification results when various numbers of 4 parameter Gaussian kernels were used. With 6 kernels the test set classification rates for the seascape data

far exceeds many of the nonlinear classifier results using fixed features, given in Chapter 4. However, only about a 1% increase in classification has been achieved by doubling the number of parameters per kernel.

Number of kernels, $M$	Classification %	T
3	89.0 (1.2)	24
6	91.0 (1.3)	45
9	91.5 (1.0)	66
12	91.75 (1.4)	87
15	92.25 (1.1)	108

**Table 5–6:** Seascape: Adapting Gaussian kernel positions and widths. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests. The value T represents the total number of parameters in the model.

The super-kernels for a 9 kernel model are shown in Figure 5–11 for the seascape data. The sailboats again were identified by the existence of the top of the mast, the motor boats by the horizontal ends, and the buoys by strong central thermal activity. More complex, or



**Figure 5–11:** Seascape:  $M = 9$  adaptive Gaussian variance and centres resulting super-kernels. Bright spot at top of sailboat kernel will heavily support the case for a sailboat classification if strong object image energy located at this point e.g. a mast. Conversely, this energy will strongly hint against the motor, and more especially, the buoy class.

subtle, feature extraction was also occurring as equivalent results were not achieved when fixed

features, generated using these simple rules, were classified. Also noticeable, in comparison with Figure 5–3, is the amount of grey areas representing regions that have little or no effect on the classification decision.

The results from the NIST data showed a slightly different trend. Adapting a larger  $P$ , low  $M$  kernel resulted in degraded performance than with the smaller  $P$  versions. This was contrary to the seascape results. Though as  $M$  increased, the benefit of the extra parameters was noticed, though less as  $M$  increased still further.

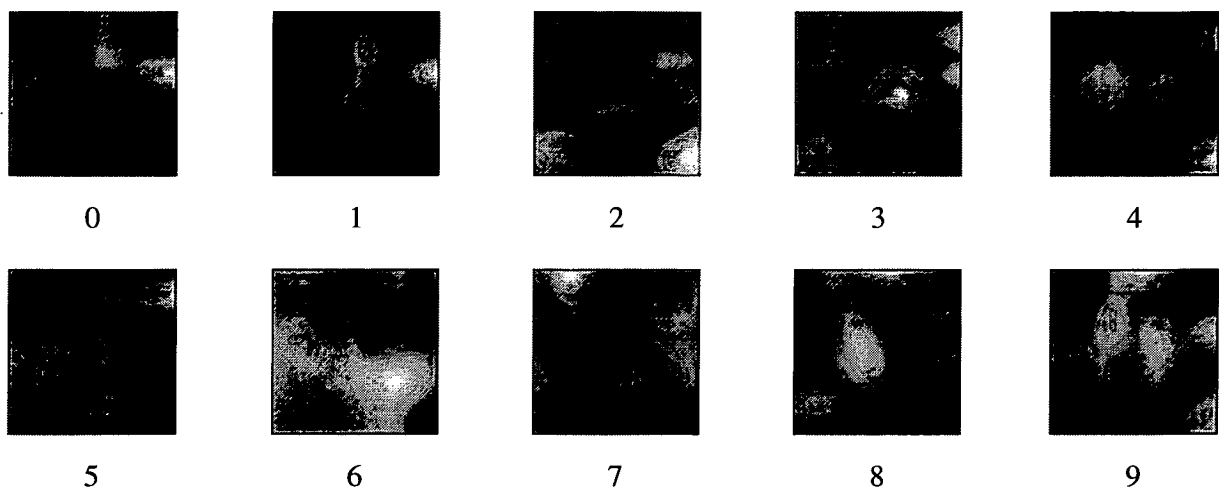
Number of kernels, $M$	Classification %	T
3	49.75 (1.3)	52
6	68.0 (2.4)	94
9	79.5 (3.0)	136
12	83.25 (2.3)	178
15	82.5 (2.9)	220

**Table 5–7:** NIST: Adapting Gaussian kernel positions and widths. Each score is the mean percentage classification over 10 different samples each consisting of 800 test vectors. The value in parentheses is the standard deviation over the 10 tests. The value T represents the total number of parameters in the model.

The super-kernels for a 15 kernel model are shown in Figure 5–12 for the NIST data. As with the previous fixed centre images definite shapes corresponding to the generalised class shape components required for classification are evident. However, ten fewer kernels were used and each kernel adapted its own position to generate these components.

The confusion matrices for a 6 kernel, 2 and 4 adaptive parameter linear model, using seascape data, are given in Table 5–8. Sailboat and buoy confusion was still the main source of error, but was reduced using the extra parameters.

From both NIST, and seascape, data experiments it was found that these 4 parameter models took longer to optimise, in terms of number of iterations. It was found that the error derivatives due to the position parameters  $(x_0, y_0)$  were much greater, in magnitude, than the equivalent derivatives for the scale parameters  $(a, b)$ . Thus, kernels tended to find positions suited for the initial scale conditions, and then slowly adapt the scale parameters. This made the performance



**Figure 5-12:** NIST:  $M = 15$  adaptive Gaussian variance and centres resulting super-kernels.

dependent on the initial scale parameters if only small training times were used. This is highlighted in Figure 5-13 for a 3 kernel model.

As the number of kernels increased the time for the kernel positions to settle, and consequently the time for completion of optimisation, increased. This is shown in Figure 5-14 for 12 kernels. It was interesting to note that after 1000 iterations some variance parameters were still changing at a rapid rate, though any improvement to classification ended after 400 iterations. One possibility was that there were too many kernels, and the associated features were redundant. Alternatively, the kernels were stopping in a regions where features were not so sensitive to changes in scale.

Guess	Correct class				Total
	Sail	Motor	Buoy		
Sail	216	1	35		252
Motor	0	164	3		167
Buoy	7	5	69		81
Total	223	170	107		500

(a)  $P=2$  (89.8% correct)

Guess	Correct class				Total
	Sail	Motor	Buoy		
Sail	203	4	21		228
Motor	1	172	1		174
Buoy	12	4	82		98
Total	216	180	104		500

(b)  $P=4$  (91.5% correct)

**Table 5-8.** Seascape: Confusion matrices for 6 kernel linear classifiers.

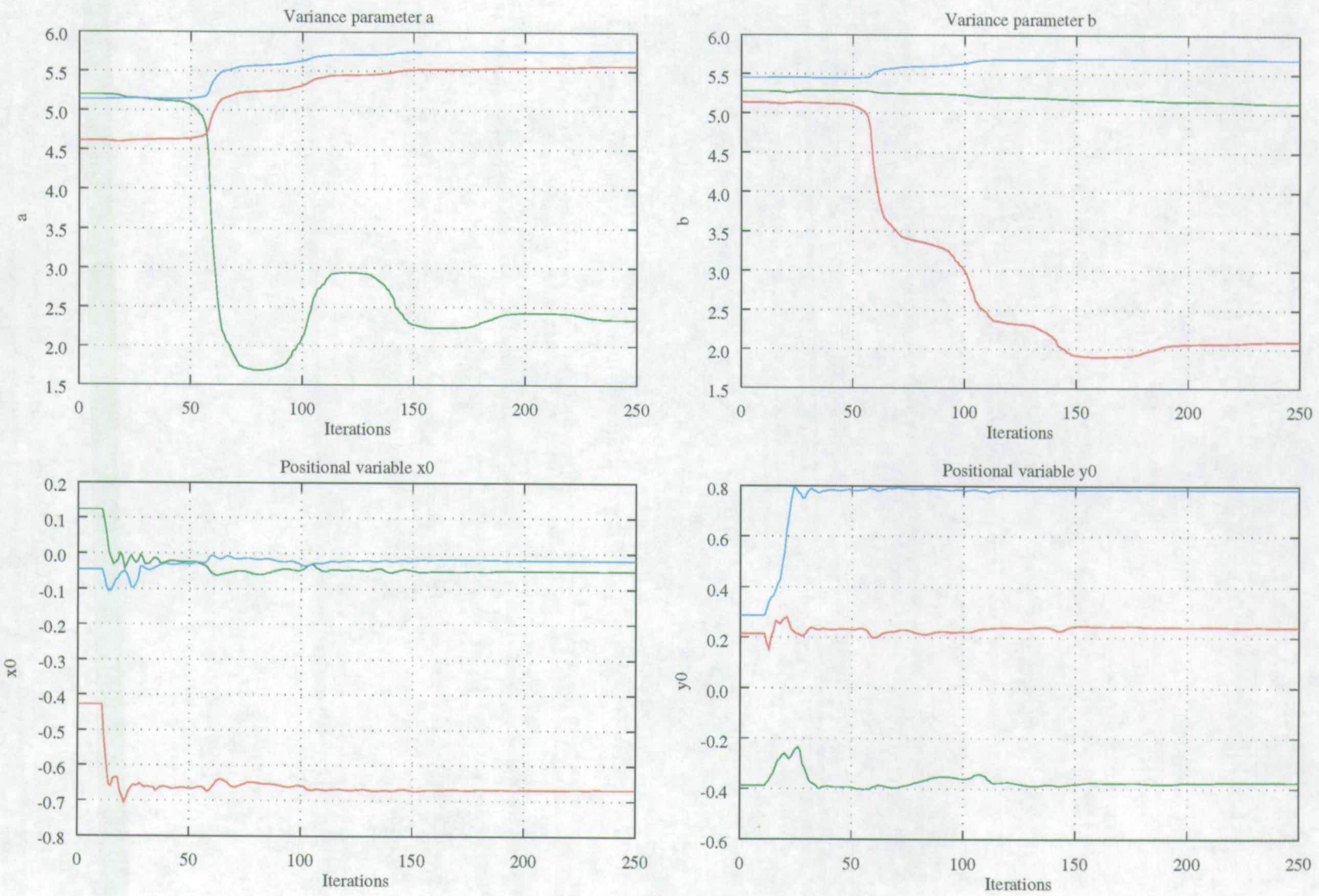


Figure 5-13: Seascape: Three kernels with  $(a, b, x_0, y_0)$  adaptive vector.



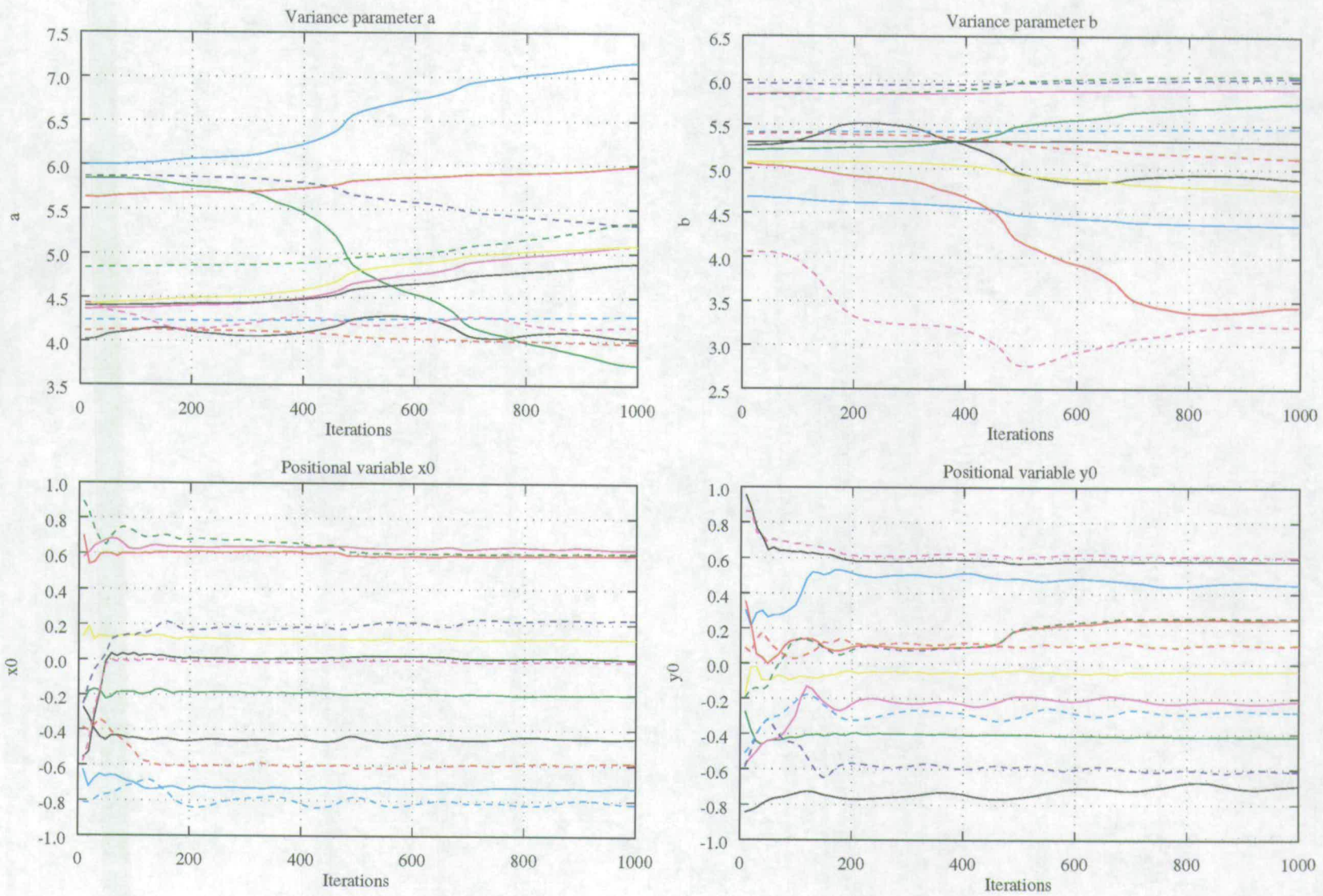


Figure 5–14: Seascape: Twelve kernels with  $(a, b, x_0, y_0)$  adaptive vector.



An adaptive wavelet

In the previous section increasing the number of parameters per kernel, only slightly increased performance for large  $M$ , but reasonable improvements were made for small  $M$ . This suggested that as  $M$  increased less flexibility in each kernel was required to achieve equivalently complex feature extractors. This seemed sensible, and a method for increasing  $P$  for small  $M$ , would be worth investigating. Allowing selective orientation of the Gaussian kernels would be one way of further increasing  $P$  for the Gaussian kernels.

Orientation encoding could be achieved by rotating  $\psi$  by  $\theta$  degrees via a simple affine transformation. The 5-dimensional parameter set would include  $(x_0, y_0, a, b, \theta)$ . It was though an opportune moment to return to the Gabor transform, a frequency modulated Gaussian. Although not strictly speaking a wavelet this kernel allowed for both orientation, and spatial frequency selection, as well as access to the usual 4 Gaussian parameters through a kernel parameter vector of  $(x_0, y_0, a, b, u, v)$ . It was also the most popular kernel used in the adaptive wavelet literature.

Experiments were applied to both the seascape and NIST data, adapting 2 or 4 of the Gabor parameters at a time but only comparative, at best, results were achieved. Finally, all six parameters were optimised together. The results are given in Tables 5–9 and 5–10.

Number of kernels, $M$	Classification %	T
3	90.5 (1.3)	30
6	91.25 (0.9)	57
9	91.5 (1.4)	84
12	92.0 (1.6)	111
15	92.25 (1.2)	138

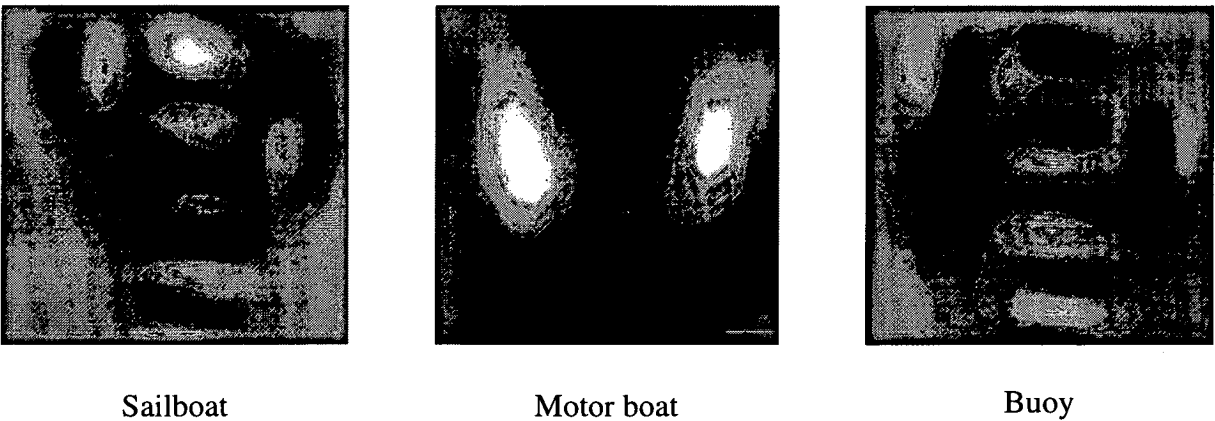
**Table 5–9:** Seascape: Adapting all 6 Gabor kernel parameters. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests. The value T represents the total number of parameters in the model.

Number of kernels, $M$	Classification %	T
3	62.0 (5.8)	58
6	74.25 (2.7)	106
9	82.5 (2.2)	154
12	83.5 (2.6)	202
15	85.75 (2.1)	250

**Table 5–10:** NIST: Adapting all 6 Gabor kernel parameters. Each score is the mean percentage classification over 10 different samples each consisting of 800 test vectors. The value in parentheses is the standard deviation over the 10 tests. The value T represents the total number of parameters in the model.

The seascape data, where only a few kernels were known to be required, soon lost the advantage of a more flexible kernel. The NIST data, where many more kernels were required, used the extra flexibility to much more effect for the lower values of  $M$ .

For completeness, the super-kernels for two 6-parameter models are shown in Figure 5–15 and Figure 5–16.



**Figure 5–15:** Seascape:  $M = 9$  adaptive 6 parameter Gabor resulting super-kernels.

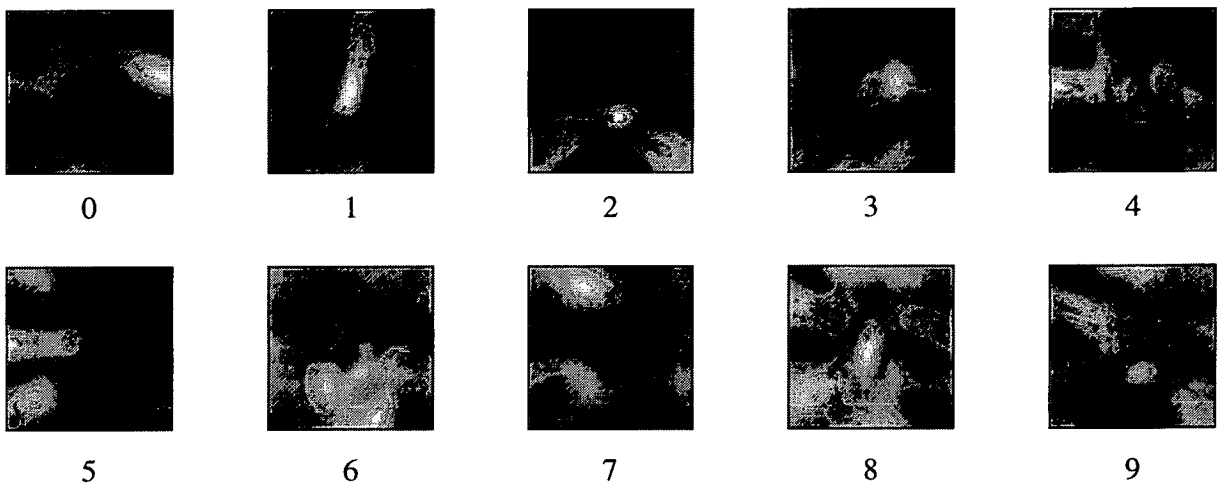


Figure 5-16: NIST:  $M = 15$  adaptive 6 parameter Gabor resulting super-kernels.

Linear conclusions

Figures 5-17 and Figure 5-18 summarise the results from this section for both seascape and NIST data.

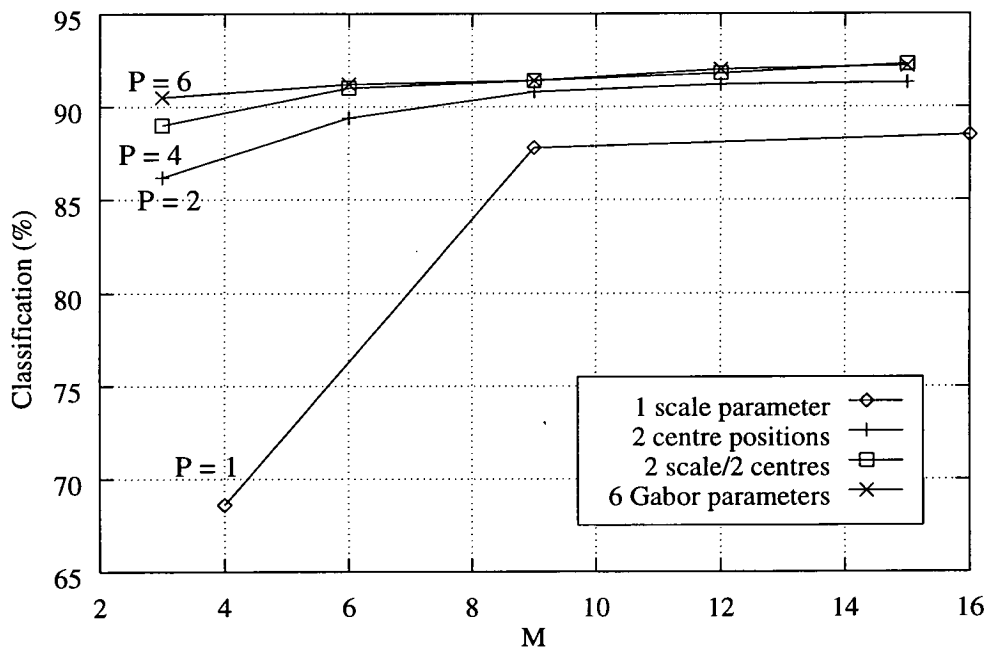
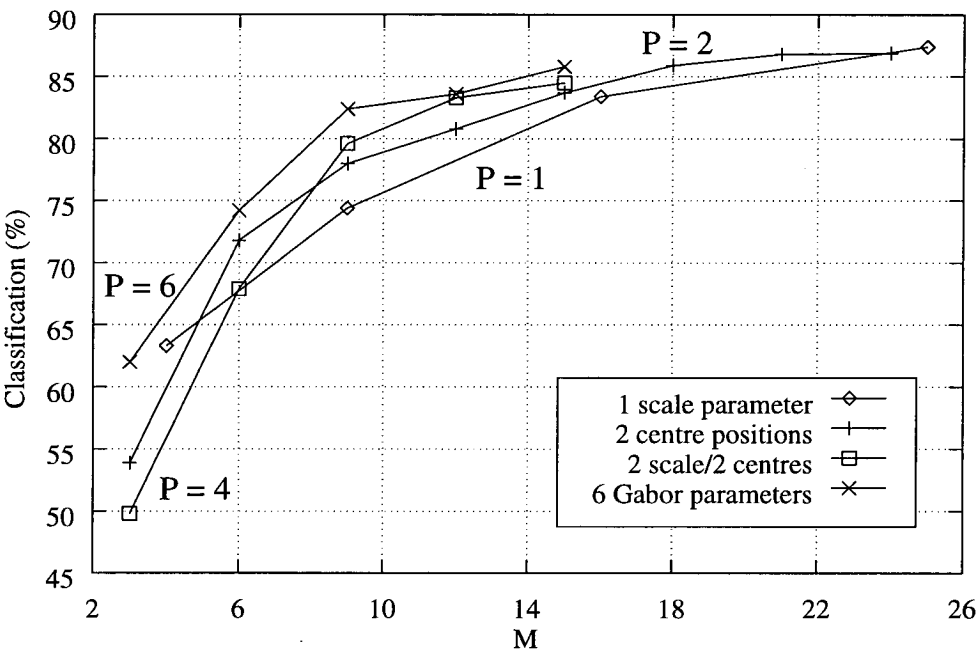


Figure 5-17: Seascape: Adaptive linear results.

The seascape data shows that for low  $M$ , a greater  $P$  yielded higher performance. As  $M$

increased the effect of  $P$  was greatly reduced. It was noticed that only a small number of kernels, ranging from 5 to 9, were required for classification.

The NIST results, at the higher  $M$ , showed a similar reduced effect of  $P$  with even the fixed centre, adaptive scale, networks showing good generalisation at  $M = 15$ , providing classification rates within a 1 or 2% of the more complicated  $P = 2, 4$  and 6 kernel models. At the lower  $M < 10$  the additional parameters appear to be detrimental to classification performance. This was due to the fact that many kernels, of any  $P$ , were required to separate the 10 NIST classes.



**Figure 5–18:** NIST: Adaptive linear results.

This suggested that it is possibly better to use a larger number of  $M$  simple kernels, as opposed to a small number of highly complex kernels, where both contain a similar number of adaptive model parameters. A conclusion also reached by researchers into kernel-based techniques for density estimation [103].

5.3.2 Nonlinear classification

The linear classification results in the previous section have shown that considerable improvements in performance were achieved compared to the linear results using fixed features described in Chapter 4. In some cases the adaptive linear model provided better performance than some of the nonlinear classifiers.

The adaptive networks were attempting to generate feature space in which objects are linearly separable. However, it was possible that no linear mapping of the image space, for either database, would have resulted in features that were completely linearly separable. Consequently, a method of nonlinearly separating an optimised feature space was required. This was performed, as stated earlier in the chapter, by extending the adaptive linear model to include a standard nonlinear layer, between the adaptive linear feature extraction and output layers. This was shown in Equation 5.3.

The extension to using the nonlinear layer was easy. The error derivatives were backpropagated through the nonlinear layer to the feature extraction kernels. The only slight problem was ensuring that the initial features generated were not so large that they saturated the outputs of the nonlinear sigmoidal units. This was achieved by careful initialisation of the the kernels and their associated weights.

The results for a model with a  $P = 4 (x_0, y_0, a, b)$  Gaussian kernel using varying numbers of kernels,  $M$ , and nonlinear units,  $N$ , are given in Table 5–11. The values are percentage mean classification rates derived from 10 tests with the standard deviation, as usual, given in brackets. Table 5–12 gives the total number of parameters in each model.

Number of kernels, $M$	Number of hidden units, $N$				
	2	4	6	8	10
3	94.0 (1.7)	94.25 (1.6)	94.5 (2.7)	95.5 (1.4)	94.75 (2.0)
6	95.75 (1.7)	96.5 (2.0)	96.0 (1.7)	96.0 (2.3)	-
9	96.75 (2.5)	96.75 (2.2)	96.5 (1.5)	-	-
12	96.5 (1.5)	97.0 (1.8)	-	-	-

**Table 5–11:** Seascape:  $M = 4$  nonlinear adaptive kernel model classification results. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests.

Number of kernels, $M$	Number of hidden units, $N$				
	2	4	6	8	10
3	29	43	57	71	85
6	47	67	87	107	-
9	65	91	117	-	-
12	83	115	-	-	-

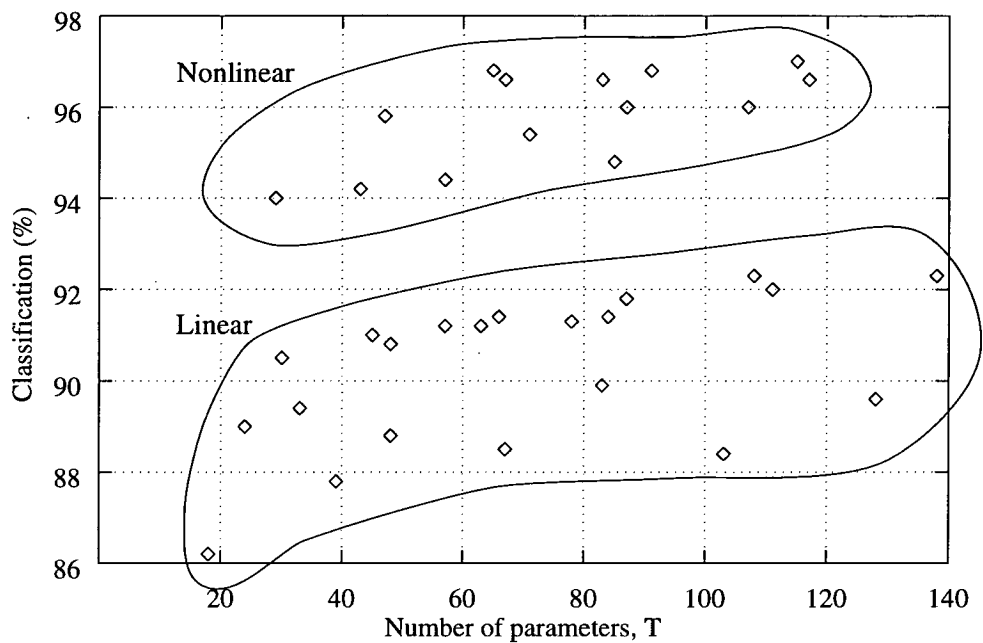
**Table 5–12:** Seascape:  $M = 4$  nonlinear adaptive kernel model parameter count.

Table 5–11 shows that for each  $M$  little, or no, improvement was made as the number of nonlinear hidden units was increased. This indicated that only a small amount of nonlinearity was required to separate the classes. However, increasing the number of adaptive kernels, for a small  $N$ , did produce improvements in classification performance. This suggested that the adaptive kernels were, in this case, flexible enough to perform the majority of the work of separating the classes such that only a relatively nonlinear discriminant was required. In other situations it is possible that the features with greatest separability will require a highly complex decision boundary. This will be indicated by a large discrepancy between the adaptive linear and adaptive nonlinear results with the respective features generated having very different distributions.

Table 5–12 shows that a model with only 67 parameters has equalled the performance of a 16 Fourier features MLP model with 8 hidden nodes (163 parameters) that scored 96% and significantly outperformed the MLP model trained with the seascape image data (4163 parameters) that scored 88.0%.

## 5.4 Review

Figure 5–19 shows how the classification results varied for both the linear and nonlinear adaptive models. A clear separation is noted between the two types of adaptive model indicating that the use of a nonlinear model was justified. Furthermore, for less than 80 parameters excellent classification results can be achieved for this database that far exceed many of the complicated feature extraction results, as well as, the highly parameterised 256 input MLP model.



**Figure 5–19:** Seascape: Classification against number of adaptive model parameters.

The use of a combined feature extraction and classification model based on adaptive kernels has been shown to be very effective in classifying images of objects derived from a real infrared seascape database. It was also shown to work well on a character recognition problem. The adaptive model itself requires no more storage than the original ATR module, in fact it has exactly the same structure. It offers ease-of-use in that no separate feature extraction and selection techniques have to be applied, as well as excellent generalisation properties. Furthermore, only a few model properties need to be adjusted to achieve good generalisation: the number of kernels, type of kernel, and for the nonlinear model the number of hidden units. The main disadvantage is that the features generated are constrained to those that can be approximated by a linear weighted summation of a fixed type of kernel.

# Invariance with adaptive kernel networks

---

The previous chapters have examined the process of feature extraction and classification, and especially how the two processes were effectively combined. The next step in the project was to incorporate various forms of *invariance* into this adaptive feature extraction and classification model. Invariance is defined here as the ability of a classifier output to remain constant regardless of certain transformations of the object, and is a fundamental requirement of a real ATR system. This chapter examines two methods that were used to introduce invariance into the adaptive model of the previous chapter.

The chapter begins with a strict definition of the term invariance, which is followed by a review of various invariant techniques, and their application to the real IR seascape problem. Finally, the chapter reports on the successes and failures that were achieved when invariance was incorporated into the adaptive model.

## 6.1 Invariance

In developing classification systems there are often constraints on the form of the mapping that links a classifiers input to its output. This *prior knowledge* can significantly aid generalisation.

One such constraint could be that the classifier outputs remain unaffected by various transformations of the input data. This is known as invariance. More formally, consider the group  $\kappa$  of transformations<sup>1</sup> acting on each of the images contained in the set  $F$ . For example,

---

<sup>1</sup>Not to be confused with the feature extraction transforms of Chapter 4.



this group may consist of all possible translations, rotations, and scalings of an image. Then, for each image classification to remain invariant, the equation

$$c(kf) = c(f) \quad \forall k \in \kappa, f \in F$$

must be true [137]. This requires constraints to be built into the design of the classifier. Specifically it is desired that

$$z(kf) = z(f) \quad \forall k \in \kappa, f \in F.$$

Subsequently, if the probability of a transform acting on an image is zero,  $P(k) = 0$ , there can be no improvement in generalisation. During the project it was assumed that all transforms,  $k \in \kappa$ , were equally probable, even if not represented by a transformed image in the database.

In ATR many types of useful invariances can be incorporated into the classification stage to improve recognition performance. These can be as simple as compensating for the time of day or image contrast, but they can be as complex as invariance against object occlusion. This project considered the geometrical distortions of translation, scaling, and rotation. These, and other distortions such as skew, can be represented by the simple affine transformation  $f(x, y) \mapsto f(x', y')$  by

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} g_1 & g_2 \\ g_3 & g_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} g_5 \\ g_6 \end{bmatrix}$$

where  $g_i$  are constants [33]. Figure 6–1 shows the effect of the three transformations on a simple structure.

Each of the three invariance were required for a specific reason. Translation invariance was needed as an object may appear at any point in the FOV. An object could also be at any distance from the sensor, and as no range data was available to compensate for this, the objects had to be classified irrespective of size. Finally, and probably most important was rotation invariance.

Rotation invariance was required to counter the equally probable effects of both rotation of the sensor and the object in the ATR environment. This may seem irrelevant to the seascape problem, in which the object images and sensor were both aligned and good classification results were achieved. This was demonstrated in Chapters 4 and 5. However, if the sensor was rotated then the classification rate was severely effected with these previous types of feature extraction

and classifier. For example, the linear, 2-parameter  $(x_0, y_0)$ , adaptive, Gaussian model with 12 kernels achieved a classification rate, on the seascape data, of 91.25% but when the sensor was artificially rotated to random orientations this dropped to 56.25%. Of course, if the sensor orientation had been known the use of suitable normalisation, or a steerable filter set, could have often, but not always, solved this problem [36]. However, there was also the converse problem of object, as opposed to sensor, rotation which could not be so easily countered.

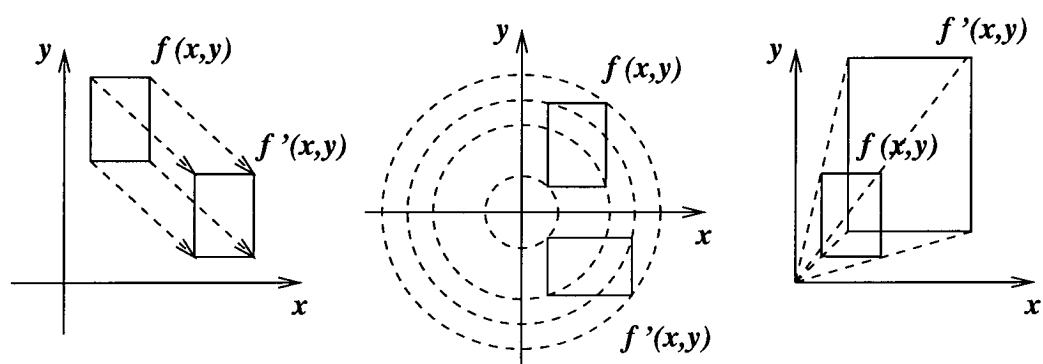


Figure 6-1: Translation, rotation, and scaling.

This project examined the feasibility of incorporating invariance into the adaptive feature extraction classifier model. Consequently, it was sensible to initially consider only simple in-plane rotations. The seascape database was not indicative of this type of object rotation but was the only non-military database available during this project. In the actual system planar rotations of the object will be prevalent and with the seascape data this was mimicked by artificial sensor rotation. The out-of-plane rotations, as discussed in Chapter 3, were treated as subclasses. Adaptive feature extraction for out-of-plane rotations is beyond the scope of this thesis.

A further point when considering invariant classification concerns discriminability. For discrimination to be possible then it is required that if  $\exists k \in \kappa$  such that  $kf_1 = kf_2$  then  $c(f_1) \neq c(f_2)$  must hold. For example, if the rotation transform of  $180^\circ$  exists in  $\kappa$  then it is impossible to discriminate between equally scaled and positioned 6's and 9's in a digit recognition system. In the seascape database, rotated versions of basic seascape shapes often looked reasonably similar. Hence, even here discriminability was a difficult problem.

Before examining the methods of invariant classification that were used to tackle these problems it is necessary to discuss a more appropriate image representation.

## 6.2 Polar image representation

In the human visual system the benefits of using polar and log-polar sampling in the retina have been discussed by several authors [127]. One main advantage is that high resolution is gained in central part of the field of view.

However, for automatic RI classification, it is simply computationally sensible to work with polar images,  $f(\rho, \theta)$ <sup>2</sup>, where  $\rho$  represents radial distance from, normally, the centre of mass and  $\theta$  is the anti-clockwise angular direction. This is because a pure rotation of a Cartesian image  $f(x, y)$  translates to a unidimensional linear shift in the  $\theta$  direction of the polar domain, i.e.  $f(\rho, \theta + \theta')$ .

Equation 6.1 demonstrates how to convert Cartesian images defined over a region  $R$  into a new domain.

$$\int_R \int f(x, y) dx dy = \int_{R^*} \int f[x(u, v), y(u, v)] |J| du dv \quad (6.1)$$

$J$  is the Jacobian,  $\partial(x, y)/\partial(u, v)$ . Hence to convert to polar coordinates let  $x = \rho \cos \theta$  and  $y = \rho \sin \theta$  and apply equation 6.1 such that

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) dx dy = \int_0^{\infty} \int_0^{2\pi} f(\rho, \theta) \rho d\rho d\theta. \quad (6.2)$$

A simple example is provided in Figure 6–2 which can be compared with the original image in Figure 3–16.

---

<sup>2</sup>Until stated continuous images will be considered, though the extension to digital images is trivial.

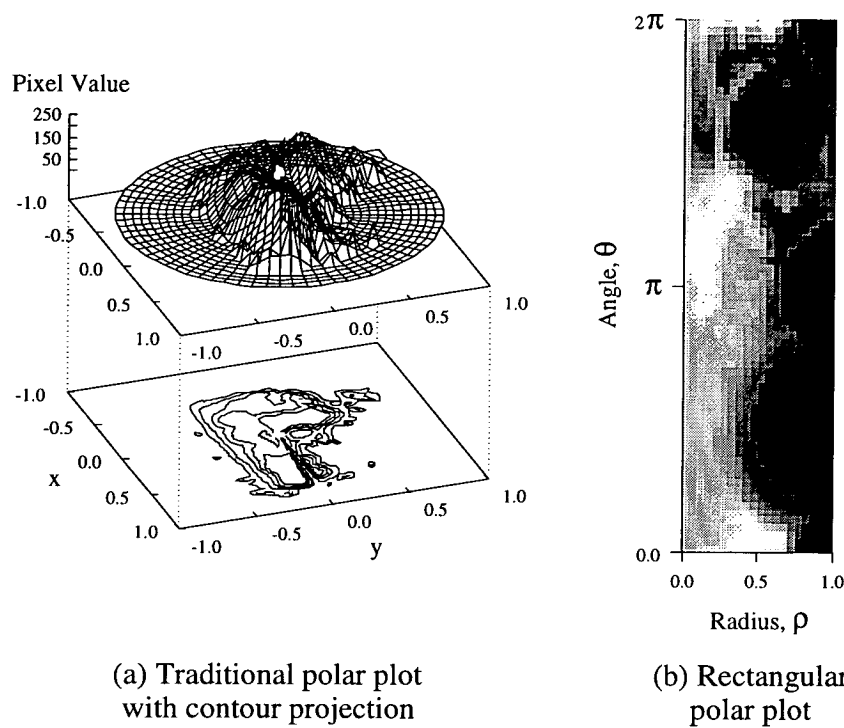


Figure 6-2: Polar plots.

### 6.3 Review of current invariant techniques

A review of invariant pattern recognition, discusses two approaches to invariant classification [137]. The first method uses invariant feature extraction followed by feature classification with, for example, a neural network. The other approach combines the two stages into a single parameterised model, usually in the context a neural network. This second idea is intuitively very appealing as classification is achieved directly against a classification error criteria. Unfortunately many of the neural-based solutions are either large and cumbersome, overparameterised, or even do not include some of the required invariances [41,44,78,99,137].

The review also raises several issues concerning invariant classification [137]. These include tolerance, discriminability as discussed before, model complexity, speed of operation, ease and speed of optimisation, generalisation ability, flexibility to new problems, and transformation retrieval. All of these are standard classification issues, except for the first and last. Tolerance considers the need for complete invariance or whether an approximation is acceptable and

transformation retrieval, for example, attempts to estimate object pose, distance from the norm, using the transformed pattern space.

In the project three methods were considered, as proposed by Barnard and Casasent [6]. These were namely:

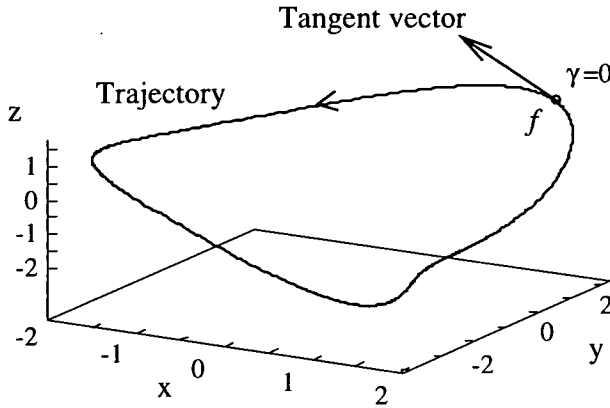
- Invariance by training or regularisation
- Invariance by structure
- Invariance through feature extraction or preprocessing

## 6.4 Invariance through training

This is a simple but brute force method of encoding classification invariance. The model is optimised using a database containing all possible transformed patterns, i.e.  $kf \forall k \in \kappa, f \in F$ . This method has several problems. The first is that  $F$  has to be very large resulting in an intensive optimisation process. Then the network is not assured to be invariant and can not extrapolate outside the patterns it has been shown. The model uses no prior knowledge of the invariance required.

A solution was proposed by Simard *et al* [113] in which a regularisation technique can be used to penalise the lack of invariance in an neural network model. The method is based around the trajectory, or manifold for more than one class of invariance, that is created when a pattern is transformed by the continuous members of the subgroup of a particular invariance. An artificial example is demonstrated in Figure 6–3, for a 3 dimensional pattern space with a single type of required invariance (e.g. rotation) parameterised by  $\gamma$ . The drawback is that an approximation, using finite differences, of the tangent vectors is required. Casasent has used a similar approach using linear piecewise approximations of feature space trajectories combined with a simple distance metric for invariant object detection in real IR images.

This approach of invariance through training was used in the project to achieve invariance to three-dimensional rotations of the seascape objects.



**Figure 6–3:** Trajectory of a transformed pattern,  $k(\gamma)f$  where  $k(0) = 1$ .

## 6.5 Invariance through structure

These are mainly the neural network techniques discussed earlier that produce large, cumbersome networks that are based on the principle of *weight sharing*. This is the constraining of specific weights to have equal values and hence encoding invariance through the structure of the model. Rumelhart *et al* used this approach for the T-C problem [92].

Examples of neural networks for invariant pattern recognition that employ weight sharing include the neocognitron, higher order neural networks, symmetric networks and time-delay neural networks [41,44,43,78,99].

The neocognitron [41] is a self-organising hierarchical multi-layer structure, as shown in Figure 6–4, that has invariance to shape distortion and partial translation invariance, and in one adaptation rotation invariance [126]. However, the hundreds of thousands of connections make it an unpractical in many ATR solutions <sup>3</sup>.

Higher order neural networks of order 3 can be used for translation, scale and rotation invariant classification. They use the idea of weight sharing amongst similar triangles in the input image. Unfortunately again they tend to produce combinatorially ( $O((N^2)^3)$ ) large

<sup>3</sup>The exception is the SAHTIRN project that uses a neocognitron classifier [25].

numbers of weights, for an  $N \times N$  image. A non-ideal solution is to use coarse coding of the input image [44,43,78].

The third main group of structural classifiers are the symmetric networks, of which Sawai's axially symmetric neural network [99] and Fukumi's first order network coin recogniser [38] are examples.

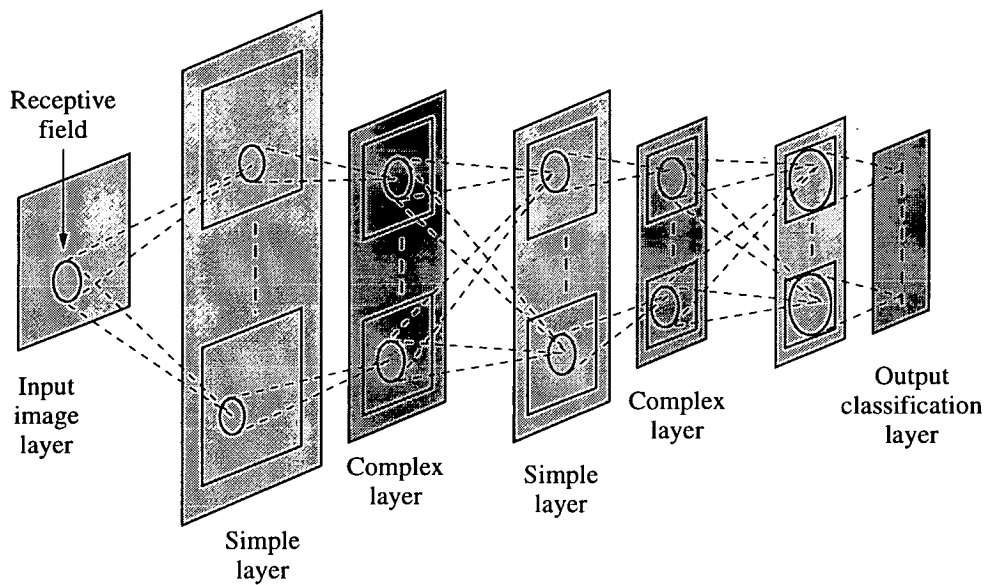


Figure 6-4: The neocognitron.

Both higher-order, and the neocognitron, were examined as possible classifiers for the project but were found to be cumbersome, difficult to optimise, and required considerably more storage for the model parameters than the existing classifier.

## 6.6 Invariance through feature extraction

The third method for invariant classification involves generating a set of features,  $d$ , from the images that are invariant under the transform group  $\kappa$  i.e.  $kd = d \forall k \in \kappa$ . This is the most popular method of all three and two specific approaches were considered for the project: preprocessing of the images such that any feature generated from the resulting image were invariant; use of an feature generating kernel,  $\psi$ , as in Chapter 5, that produced features that

were naturally invariant to the three required transformations. Each of the three transforms shall now be discussed individually.

### 6.6.1 Translation

There are two simple methods for tackling translation invariance. The first calculates the centre of mass  $(x_0, y_0)$  of a Cartesian image,  $f(x, y)$ , and shifts the origin of the coordinate system to that point i.e.  $f(x - x_0, y - y_0)$ . This is a preprocessing method, and the method of choice for the project.

The second, kernel, method uses a complex kernel of the form  $\psi(x, y) = \exp[j(ux + vy)]$  such that any arbitrary shift  $(x', y')$  in the image only produces a linear phase shift in the resulting complex feature. Hence, an invariant feature can be produced by only using magnitude data. An excellent example of this is the Fourier transform [47]. The Fourier transform,  $\mathcal{F}$  of an image  $f(x, y)$  is given by

$$\mathcal{F}_{u,v}\{f(x, y)\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy \quad (6.3)$$

and for a shifted version of the same image

$$\mathcal{F}'_{u,v}\{f(x - x', y - y')\} = \mathcal{F}_{u,v}\{f(x, y)\} e^{-j2\pi(ux'+vy')}. \quad (6.4)$$

Thus the feature,  $|\mathcal{F}_{u,v}\{f(x, y)\}|$ , is invariant to translations. However, this assumes that the majority of discriminatory information is contained in the magnitude [129].

### 6.6.2 Scale

In a similar dual manner the problem of scale invariance was approached. However, the tendency was to perform preprocessing initially. Chapter 3 described how the seascape objects were resampled to a standard size. The main reason for this was to allow the direct application of the object to classification or feature extraction systems. However, it also introduced basic scaling invariance. Another method was to normalise the image such that the average radius, from the centre of mass, of the object was identical for each object. Similarly, the distance



from the centre to the extremity of the object could be scaled to unity. This last method was found to be less robust and easily effected by poor segmentation.

Another very popular, and biologically plausible approach, was the use of the log-polar transform. This has already been used in ATR systems [15]. Section 6.2 showed how to convert a Cartesian image,  $f(x, y)$ , into the polar domain,  $f(\rho, \theta)$ . Scale invariance becomes possible if in the polar transform the identity  $\rho = e^r$  is used to generate a new image,  $f'(r, \theta) = f(e^r, \theta)$ . Scaling the original image by a factor  $\beta$  then produces a linear shift in the log-polar image of  $\ln \beta$  i.e.  $f'(r + \ln \beta, \theta)$ . Shifts such as these can be easily countered by either of the translation invariant techniques discussed in the previous section. This was found to be a very successful with the seascape data.

Low order image moments have also been successfully applied to the problem of scale invariance and will be explained in the following section, in conjunction with their rotation invariance properties.

### 6.6.3 Rotation

For a classification system to be rotation invariant (RI) the condition

$$z(f(\rho, \theta)) = z(f(\rho, \theta + \theta')) \forall \theta'$$

must hold true. There have been many approaches taken to RI classification [6,137,101,39,104,129]; some preprocessing based and others complex kernel based.

#### Preprocessing

Some of the preprocessing methods for RI feature extraction are intuitive, such as using eigenvector analysis to rotate the image such that the directions of maximum variance align (unfortunately there are two directions of maximum variance), and some are more complicated, like Fourier descriptors [80] which analyse the spectrum of boundary contours of an object, as seen in Chapter 3. One simple method, named  $\theta$  normalisation was used in the project, calculates the mean of a polar, or log-polar, image in the  $\theta$  direction. This was very appealing as the log-polar image could be easily combined, for little extra computation, with the scale

invariant generating process of using the  $\ln\rho$  directional image mean, as in the previous section. However, unlike the  $\ln\rho$  mean the  $\theta$  mean could not be calculated so easily as the polar images were periodic in the  $\theta$  direction. This was solved using the circular mean [71].

A polar image,  $f(\rho, \theta)$ , has a circular mean,  $\theta_0 = \Gamma(f)$ , given by

$$\cos \theta_0 = \frac{C(\theta)}{R(\theta)}$$

or, alternatively, by

$$\sin \theta_0 = \frac{S(\theta)}{R(\theta)}$$

where

$$C(\theta) = \int_0^1 \int_0^{2\pi} \cos \theta f(\rho, \theta) d\theta d\rho, \quad S(\theta) = \int_0^1 \int_0^{2\pi} \sin \theta f(\rho, \theta) d\theta d\rho, \quad (6.5)$$

and

$$R(\theta) = \sqrt{C^2(\theta) + S^2(\theta)}.$$

Invariance is achieved by shifting  $f(\rho, \theta)$  by  $\theta_0$  generating a RI image,  $f(\rho, \theta + \theta_0)$ .

**Proof:** Consider an image  $f(\rho, \theta)$  and a rotated version of the same image shifted by  $\theta'$  in the  $\theta$  direction,  $f(\rho, \theta + \theta')$ . Let

$$\theta_0^a = \Gamma(f(\rho, \theta)) \quad \text{and} \quad \theta_0^b = \Gamma(f(\rho, \theta + \theta')),$$

and then for circular mean normalisation to be rotation invariant the following condition must hold:

$$f(\rho, \theta + \theta_0^a) = f(\rho, \theta + \theta' + \theta_0^b), \quad \text{or} \quad \theta_0^b = \theta_0^a - \theta'.$$

So, for the rotated image  $f(\rho, \theta + \theta')$

$$C^2(\theta) = \left[ \int_0^1 \int_0^{2\pi} \cos \theta f(\rho, \theta + \theta') d\theta d\rho \right]^2$$

and by letting  $\Theta = \theta + \theta'$  it can be seen that

$$\begin{aligned}
 C^2(\Theta) &= \left[ \int_0^1 \int_{\theta'}^{2\pi+\theta'} \cos(\Theta - \theta') f(\rho, \Theta) d\Theta \rho d\rho \right]^2 \\
 &= \left[ \cos \theta' \int_0^1 \int_{\theta'}^{2\pi+\theta'} \cos \Theta f(\rho, \Theta) d\Theta \rho d\rho + \sin \theta' \int_0^1 \int_{\theta'}^{2\pi+\theta'} \sin \Theta f(\rho, \Theta) d\Theta \rho d\rho \right]^2 \\
 &= \cos^2 \theta' \left[ \int_0^1 \int_{\theta'}^{2\pi+\theta'} \cos \Theta f(\rho, \Theta) d\Theta \rho d\rho \right]^2 + \sin^2 \theta' \left[ \int_0^1 \int_{\theta'}^{2\pi+\theta'} \sin \Theta f(\rho, \Theta) d\Theta \rho d\rho \right]^2 \\
 &\quad + 2 \cos \theta' \sin \theta' \left[ \int_0^1 \int_{\theta'}^{2\pi+\theta'} \cos \Theta f(\rho, \Theta) d\Theta \rho d\rho \right] \left[ \int_0^1 \int_{\theta'}^{2\pi+\theta'} \sin \Theta f(\rho, \Theta) d\Theta \rho d\rho \right]
 \end{aligned}$$

Similarly,

$$\begin{aligned}
 S^2(\Theta) &= \cos^2 \theta' \left[ \int_0^1 \int_{\theta'}^{2\pi+\theta'} \sin \Theta f(\rho, \Theta) d\Theta \rho d\rho \right]^2 + \sin^2 \theta' \left[ \int_0^1 \int_{\theta'}^{2\pi+\theta'} \cos \Theta f(\rho, \Theta) d\Theta \rho d\rho \right]^2 \\
 &\quad - 2 \cos \theta' \sin \theta' \left[ \int_0^1 \int_{\theta'}^{2\pi+\theta'} \cos \Theta f(\rho, \Theta) d\Theta \rho d\rho \right] \left[ \int_0^1 \int_{\theta'}^{2\pi+\theta'} \sin \Theta f(\rho, \Theta) d\Theta \rho d\rho \right]
 \end{aligned}$$

Now  $R^2(\Theta) = C^2(\Theta) + S^2(\Theta)$  such that

$$\begin{aligned}
 R^2(\Theta) &= \left[ \int_0^1 \int_{\theta'}^{2\pi+\theta'} \cos \Theta f(\rho, \Theta) d\Theta \rho d\rho \right]^2 + \left[ \int_0^1 \int_{\theta'}^{2\pi+\theta'} \sin \Theta f(\rho, \Theta) d\Theta \rho d\rho \right]^2 \\
 &= \left[ \int_0^1 \int_0^{2\pi} \cos \Theta f(\rho, \Theta) d\Theta \rho d\rho \right]^2 + \left[ \int_0^1 \int_0^{2\pi} \sin \Theta f(\rho, \Theta) d\Theta \rho d\rho \right]^2 = R^2(\theta) \quad (6.6)
 \end{aligned}$$

So  $R(\theta)$  is invariant to changes in image rotation but what about  $C(\theta)$ ?

$$\begin{aligned}
 C(\Theta) &= \int_0^1 \int_{\theta'}^{2\pi+\theta'} \cos(\Theta - \theta') f(\rho, \Theta) d\Theta \rho d\rho \\
 &= \cos \theta' \int_0^1 \int_0^{2\pi} \cos \Theta f(\rho, \Theta) d\Theta \rho d\rho + \sin \theta' \int_0^1 \int_0^{2\pi} \sin \Theta f(\rho, \Theta) d\Theta \rho d\rho
 \end{aligned}$$

Combining this with  $R(\Theta)$  from equation 6.6 and letting  $\Theta = \theta$  it can be seen that

$$\cos \theta_0^b = \frac{C(\Theta)}{R(\Theta)} = \cos \theta' \cos \theta_0^a + \sin \theta' \sin \theta_0^a = \cos(\theta_0^a - \theta')$$

such that

$$\theta_0^b = \theta_0^a - \theta'$$

and therefore circular mean normalised images are rotationally invariant.

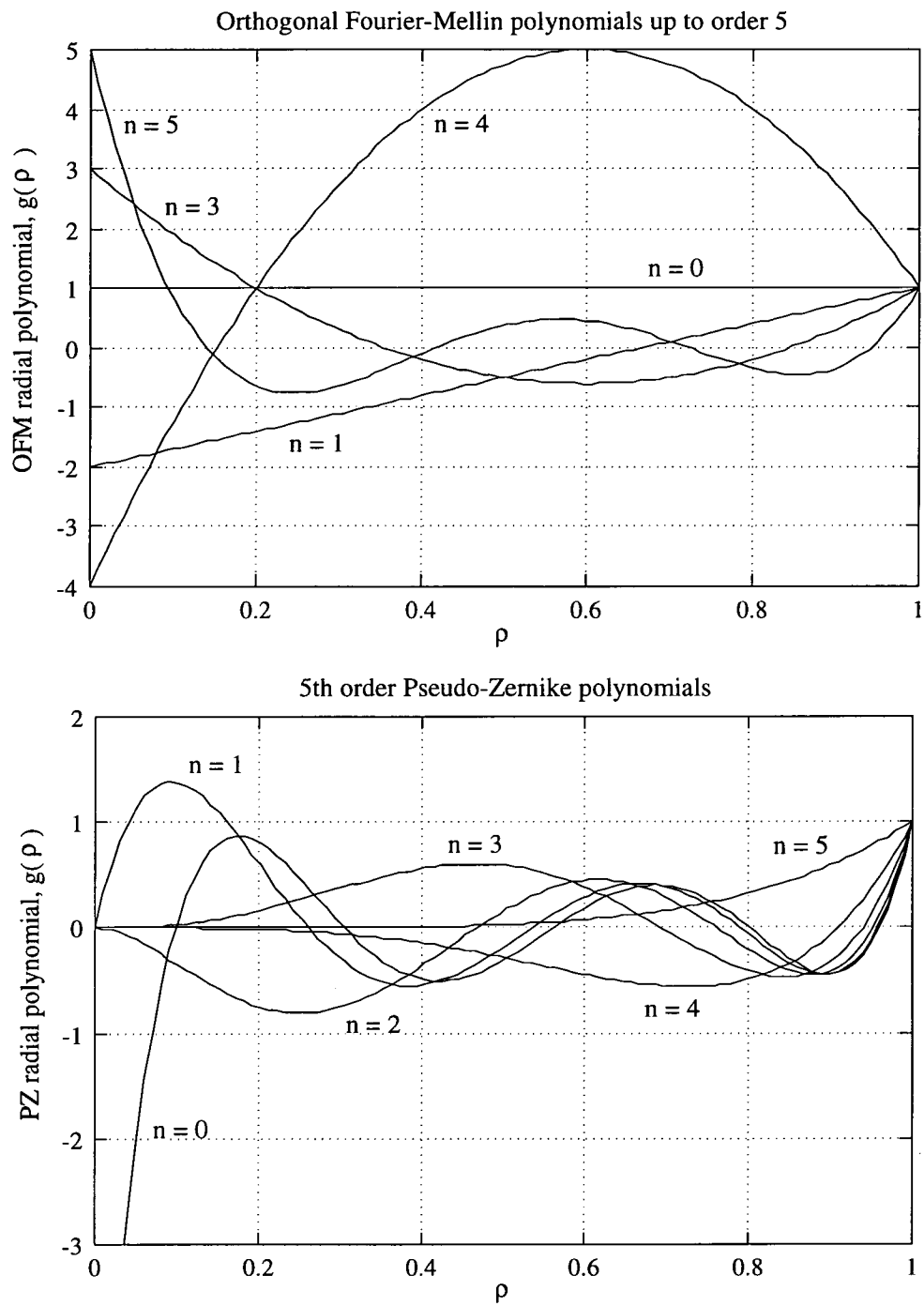
### Complex kernel feature extraction

Another method of generating a set of RI features was to use a kernel such that all points in the kernel equidistant from the centre had the same value. This approach was taken by Fukumi in his shared weight, neural network, coin recognition system [38]. Unfortunately at a particular distance,  $\rho'$ , from the centre of the image,  $f(\rho', \theta)$  can have any arbitrary arrangement provided that the sum over all  $\theta$  remained constant. The features can be expressed as

$$d_i = \int_0^1 \int_0^{2\pi} f(\rho, \theta) \psi_i(\rho, \theta) d\theta \rho d\rho \quad (6.7)$$

where  $\psi_i(\rho, \theta) = g(\rho)$ , a radial function or polynomial.

This simplistic approach can be extended to take into account variations of the image in the  $\theta$  direction by using complex kernels. In their paper concerning Zernike circular polynomials Bhatia and Wolf [11] demonstrated that for a kernel to provide RI about the centre of mass of an object it must be of the form  $g(\rho) \exp(jm\theta)$  where  $m$  represents circular harmonic order and, as stated before,  $g(\rho)$  is a radial polynomial. Examples of radial polynomials are provided in Figure 6–5.



**Figure 6–5:** Examples of two different classes of radial polynomials,  $g(\rho)$ .

These complex kernels can be used to generate RI features,  $d_i$ , as in Equation 6.8 where  $*$  denotes the complex conjugate and  $| \cdot |$  complex magnitude.

$$d_i = \left| \int_0^1 \int_0^{2\pi} f(\rho, \theta) \psi_i^*(\rho, \theta) d\theta \rho d\rho \right| \tag{6.8}$$

The ability of this transform to achieve RI is easily demonstrated. Each RI feature,  $d^a$ , is determined by the equation

$$d^a = \left| \int_0^1 \int_0^{2\pi} f(\rho, \theta) \psi^*(\rho, \theta) d\theta \rho d\rho \right| = \left| \int_0^1 \int_0^{2\pi} f(\rho, \theta) g(\rho) e^{-jm\theta} d\theta \rho d\rho \right|.$$

To prove the features are RI the effect of rotating the image by  $\theta'$  is compared with the new feature,  $d^b$ , given by

$$d^b = \left| \int_0^1 \int_0^{2\pi} f(\rho, \theta + \theta') g(\rho) e^{-jm\theta} d\theta \rho d\rho \right|$$

and by letting  $\Theta = \theta + \theta'$  and knowing that  $| \exp(jm\theta') | = 1$

$$d^b = \left| e^{jm\theta'} \int_0^1 \int_0^{2\pi} f(\rho, \Theta) g(\rho) e^{-jm\Theta} d\Theta \rho d\rho \right| = d^a.$$

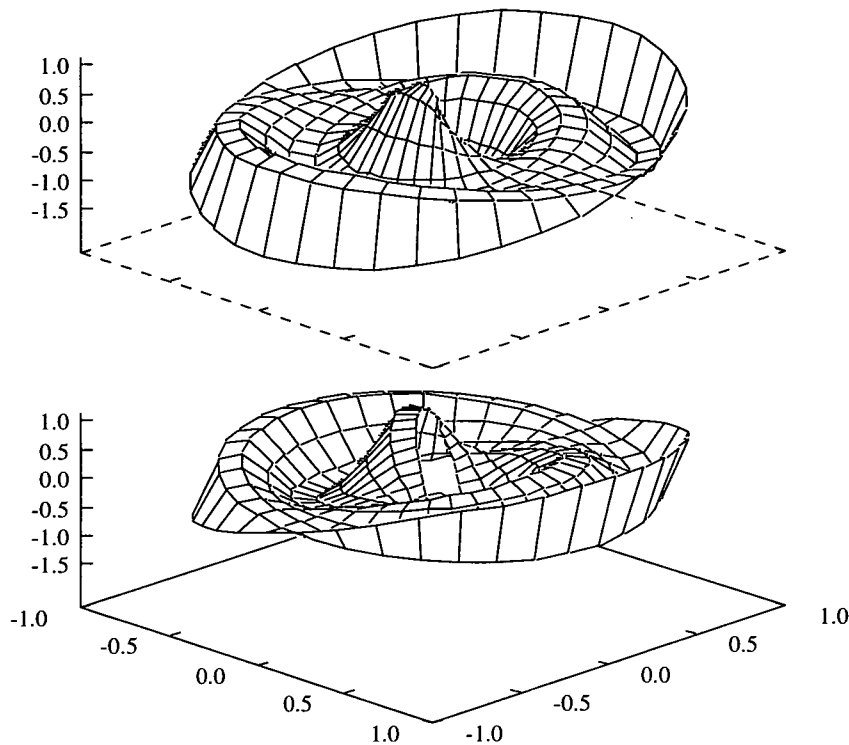
The choice of  $g(\rho)$  and  $m$ , which determine the shape of the kernel, are crucial for an acceptable classification rate and four types of kernel derived from Fourier-Mellin (FM), orthogonal Fourier-Mellin (OFM), Zernike (ZE) and pseudo-Zernike (PZ) moments have been found to work well [109,124,107,106,105,61,62,108,67]. Fourier-Mellin moments,  $M_{im}$ , use the kernel  $\psi_i(\rho, \theta) = \rho^i \exp(jm\theta)$  where in this thesis  $i$  is integer valued. Sheng and Shen [109] derived OFM moments by orthogonalisation of the sequence  $1, \rho, \rho^2, \dots, \rho^n$ . This generated a set of orthogonal  $g(\rho)$  such that  $\psi_{in}(\rho, \theta) = \exp(jm\theta) \sum_{s=0}^n \beta_{ins} \rho^s$ . Two other sets of moments were discovered by a similar orthogonalisation of the sequences  $\rho^{|m|}, \rho^{|m|+2}, \dots, \rho^{|n|}$  and  $\rho^{|m|}, \rho^{|m|+1}, \dots, \rho^{|n|}$ . These are the ZE and PZ moments respectively [11]. The real and imaginary parts of one ZE kernel is given in Figure 6–6. In the same way as the OFM, the ZE and PZ kernels can be expressed as linear combinations of weighted natural powers of  $\rho$  but with  $\beta_{ins} = 0$  for  $s < m$ . More generally,

$$d_i = \left| \sum_{s=0}^n \beta_{ins} \int_0^1 \int_0^{2\pi} f(\rho, \theta) \rho^s e^{-jm\theta} d\theta \rho d\rho \right| = \left| \sum_{s=0}^n \beta_{ins} M_{sm} \right| \quad (6.9)$$

whereby suitable choice of  $\beta_{ins}$  can generate any of the required moments.

Teh and Chin [124] tested various image moments for information redundancy, noise sensitivity and image reconstruction capability. Of the moments examined Zernike had the

best overall performance. However, the position of the ZE  $g(\rho)$  zeros, than say those of OFM, might not be so suitable for certain types of RI classification [109].



**Figure 6–6:** The real and imaginary kernels of one Zernike kernel.

An excellent introduction to moment-based features is given by Teague, and others [122, 1,124,62,9]. Moments have been successfully applied to applications such as ship, plane, and character recognition, as well as being incorporated into many ATR solutions [117,31,61].

**Other techniques**

There are other techniques for RI classification which shall be mentioned for completeness. Hu introduced a set of algebraic moments based on nonlinear combinations of normalised regular moments [57]. These translation, rotation, and scale invariant features were based on Cayley-Sylvester’s theory of algebraic invariants and corrected by Reiss [88]. All these moment-based techniques are a generalisation of a basic theory of moments which are a general class of invariants.

The Fourier transform can also provide RI classification. This can be seen by noting that a rotation in the image plane results in a similar rotation of the Fourier plane. The polar Fourier transform is given as

$$\mathcal{F}_{r,t}\{f(\rho, \theta)\} = \int_0^\infty \int_0^{2\pi} f(\rho, \theta) e^{-j2\pi\rho r \cos(\theta-t)} d\theta \rho d\rho \quad (6.10)$$

and a linear shift in  $\theta$  by  $\theta'$  radians results in an equivalent linear shift in  $t$  such that then  $\mathcal{F}_{r,t}\{f(\rho, \theta - \theta')\} = \mathcal{F}_{r,t+\theta'}\{f(\rho, \theta)\}$ . Rotation invariance is achieved by binning the Fourier plane into radial bins. Conversely, binning into wedges provides scale invariance. This is known as the wedge-ring feature extractor.

Other various methods for RI include features based on the grey level histogram of the objects, as in Chapter 3, and using simple descriptive measures as features. Also fractional central moments [55] and constraint-based approaches [6] which transform images along feature trajectories until a set of constraints,  $C_i$ , is satisfied such that  $C_i(\cdot) = 0 \forall i$  which is known as the constraint surface. A simple example is the up-righting of alphanumeric characters to the horizontal.

The final, and a special case, form of RI classification are the RI matched filters such as circular harmonic filters. These use the decomposition of images into a series orthogonal basis images. An image  $f(\rho, \theta)$  may be written as the angular Fourier series

$$f(\rho, \theta) = \sum_{m=-\infty}^{\infty} f_m(\rho) \exp(jm\theta) \quad (6.11)$$

The angular-Fourier series coefficients  $f_m(\rho)$  are called the angular harmonics and are given by

$$f_m(\rho) = 1/2\pi \int_0^{2\pi} f(\rho, \theta) \exp(-jm\theta) d\theta \quad (6.12)$$

and the energy associated with each angular harmonic is

$$E_m = 2\pi \int_0^\infty |f_m(\rho)|^2 \rho d\rho. \quad (6.13)$$

In this way a filter set can be constructed to perform RI classification. It is without the scope of this thesis to consider these filters any further.



## 6.7 Digital approximation

In the previous sections invariance has been discussed with respect to a continuous image. It must be noted that these features are strictly only invariant when computing for a continuous image. Consequently, there were effects introduced when replacing the continuous integrals by the digital approximation of summations in the digital ATR system. These were due to sampling, digitising, and quantising of the original scene. Teh and Chin investigated the effects of digital approximations of moment invariants [123]. There was no time in the project to examine the effects on the seascape data.

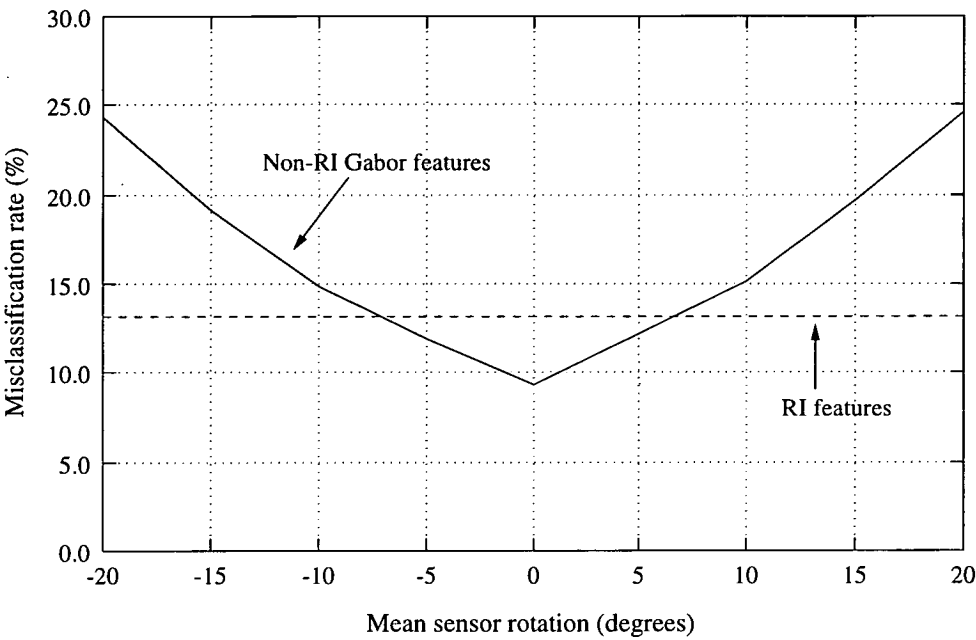
## 6.8 Classification of fixed RI features

In Section 6.3 three approaches to invariant classification were discussed. Of these three invariance through feature extraction or preprocessing were described as the most popular. This section provides results for the rotation invariant classification performed on the seascape database. As described previously, this database did not naturally contain objects with in-plane rotation. The rotation was introduced by artificial sensor rotation.

The first two experiments were designed to demonstrate how classification rates of some of the non-RI features discussed in Chapter 4 would be effected by small deviations away from the upright (*tolerance*) and how they would alter with objects of random rotations.

Figure 6–7 shows how a set of fixed Gabor features coped when small, random sensor rotations were introduced on the seascape object database. For zero-degree added rotation the Gabor features were identical to that reported in Chapter 4. RI-features, typically, performed worse, in this situation as non-RI features have additional information. However, as sensor rotation was increased the misclassification rate of the non-RI features increased sharply. At a mere 20° the classification rate dropped by 10%. This meant a system based on the assumption of upright objects must perform the alignment of the objects accurately.

The next set of results, given in Table 6–1, showed how three different features ( Gabor



**Figure 6–7:** Seascape: Effect on non-RI feature classification by small sensor rotations.

chosen by BaB, Fourier chosen by Wilks’ score, and zoning ) coped with a seascape database in which all objects were randomly rotated between 0° and 360°. They were classified using a linear and 7-NN classifier.

Feature	#	Unrotated (%)		Rotated (%)		Difference (%)	
		Linear	7-NN	Linear	7-NN	Linear	7-NN
Gabor	8	83.5	91.25	43.5	47.0	40.0	44.25
Fourier	16	86.25	96.0	46.75	49.75	39.5	46.25
Zone	16	86.0	92.5	41.0	42.25	44.75	50.0

**Table 6–1:** Seascape: Non-RI features with a rotated database. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests.

As can be in Table 6–1 there was a considerable reduction in the classification rate. This was unacceptable. The classification rate must remain unchanged by simple rotations of the

object. Several RI techniques, that were discussed earlier in this chapter, were applied to the same rotated object database. These RI features included the popular moment-based features chosen by the BaB algorithm, and standard features applied to  $\theta$ -normalised images <sup>4</sup>. As can be seen in Table 6–2 significant improvements were achieved. Table 6–3 shows the new classification confusion matrices.

Feature	#	Linear (%)	7-NN (%)
<b>Moments:</b>			
Hu	7	64.5 (2.0)	73.25 (1.9)
Fourier-Mellin (FM)	15	76.0 (1.7)	81.0 (2.1)
Orthogonal FM (OFM)	15	75.0 (1.8)	82.75 (2.0)
Pseudo-Zernike	15	75.0 (1.6)	83.5 (1.8)
Complex	15	74.5 (1.4)	78. 25(1.9)
<b><math>\theta</math>-normalised:</b>			
Gabor	15	66.5 (2.6)	72.0 (2.3)
Gaussian	16	63.0 (2.0)	73.25 (2.1)
Geometrical	15	63.75 (2.7)	74.25 (2.4)

**Table 6–2:** Seascape: RI features with a rotated database. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests.

As with the results provided in Chapter 4, Table 6–2 provided a benchmark with which to compare any new adaptive results. Certain types of moment performed effectively, and only a drop of approximately 5% in classification was recorded over the non-RI features. The  $\theta$ -normalised features performed poorly now though, indicating in the new images, the positions the shapes of the feature extracting kernels were incorrect. Table 6–3 shows that when the RI features were considered the source of confusion also altered; the problem was now distinguishing sailboats and motor boats.

<sup>4</sup>Fourier features were not calculated due to the non-square nature of the new images.

Guess	Correct class				Total
	Sail	Motor	Buoy		
Sail	166	19	5		190
Motor	7	97	12		110
Buoy	7	13	80		100
Total	180	123	97		400

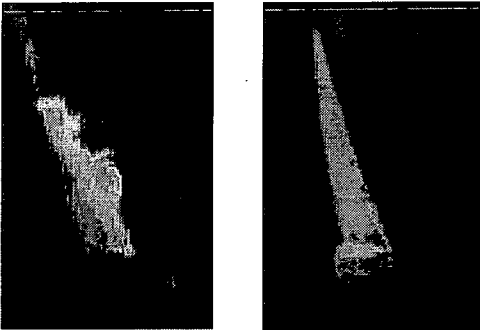
Guess	Correct class				Total
	Sail	Motor	Buoy		
Sail	131	11	5		147
Motor	30	90	23		143
Buoy	19	22	69		110
Total	180	123	97		400

(a) Pseudo-Zernike (84.25% correct)

(b) Gaussian (72.5% correct)

**Table 6–3:** Seascape: 7-NN classifier confusion matrices.

It was found that at certain angles a rotated motor boat had many similarities with a sailboat. This is demonstrated in Figure 6–8.



**Figure 6–8:** Seascape: The motor boat, on the left, has been rotated by 80 counterclockwise.

### 6.9 Adaptive invariant techniques

The previous section has shown how the RI moment-based classifiers were successful in discriminating between the seascape objects. Thus, it seemed sensible to attempt to include the moment kernels into a combined feature extraction and classification model. In this way the moment parameters could be adapted to provide improved classification.

6.9.1 Adaptive complex kernel feature extraction

Previously, it was stated that the choice of the moment radial polynomial,  $g(\rho)$ , and circular harmonic order,  $m$ , control the classification rate, as they control the shape of the moment kernels, and consequently the RI features. Thus, many different types of kernel have been proposed including Fourier-Mellin, orthogonal Fourier-Mellin, Zernike and pseudo-Zernike moments which have been found to work well [109,124]. However, these have not always been devised for image recognition systems. A method where  $g(\rho)$ , at least, for a particular problem could be identified automatically would be very beneficial.

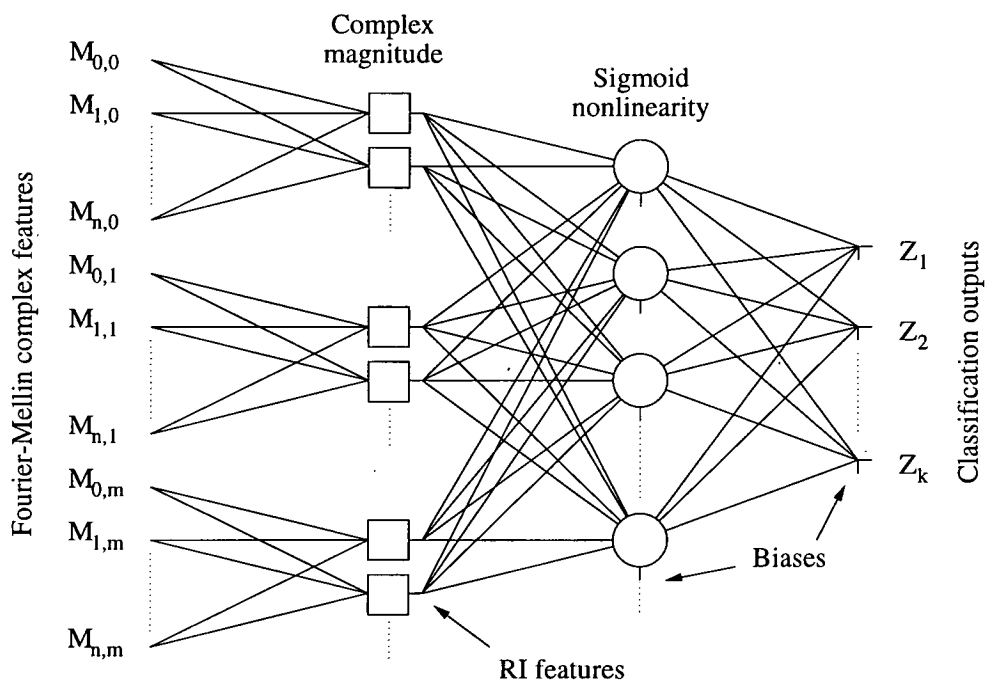


Figure 6–9: The adaptive complex kernel classifier model.

In order to combine the feature extraction into an overall classification model the values  $\beta_{ins}$ , from Equation 6.9, were incorporated as classification parameters to be optimised [116]. This would allow for the automatic selection of a suitable set of  $g(\rho)$ 's for a particular object recognition task. The RI kernel feature extraction can be visualised, as in Figure 6–9, as an extra preprocessing layer in a 2 layer MLP model, with the first layer containing complex magnitude nonlinearities, and weights as  $\beta_{ins}$ . As stated earlier, by fixing the  $\beta_{ins}$  weights, all the types of moment discussed could be generated by the model. But, by adapting the weights using the classification error improved classification was hoped to be achieved.

There were several problems associated with this technique. First, imagine the simple problem of optimising, with respect to a sum-of-squares error criterion, the network  $z = w \mid \beta M \mid$  where  $w$  is the output weight and  $\beta$  and  $M$ , the Fourier-Mellin moments, are as in Equation 6.9. The error surface for this problem, using FM features derived from two distinct classes of simple rotated images is shown in Figure 6–10. The solution requires a positive output weight. However, if the network is optimised, starting with a negative output weight, the shape of the error surface in the negative region can cause line searching optimisation techniques to fail. Also, with a positive output weight the network can be expressed as  $z = \mid w \beta M \mid$  and hence there is ill-conditioning.

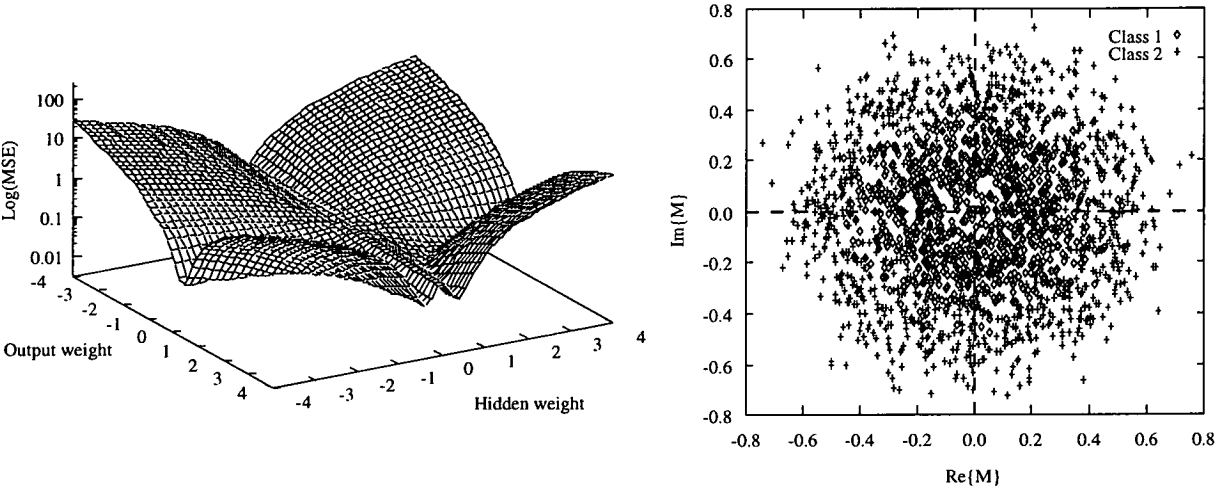


Figure 6–10: Mean squared error (MSE) surface for a simple problem.

Fortunately, a non-derivative based optimisation method, known as simplex (see Appendix A), was able to provide a working alternative and results for the seascape data were able to be recorded. At this point a second problem was noted. As seen in Figure 6–9 there could be no interaction between features and FM inputs of different  $m$  for RI to be maintained. This introduced an unwanted complexity, and also required consideration of  $n$  and  $m$ , as well as the requirement for significant numbers of inputs.

The first test with the seascape database used FM inputs with fixed  $m = 2$ , 8 sigmoid units; but varying number of generated RI features. The test was run for 10,000 epochs. The  $\beta$  weights were either fixed to generate particular types of moments or were adaptable. Ten FM ( $m = 2$ ) complex features were used as inputs. The classification results are given in Figure 6–11. The classification rate peaked at 73.5% using 6 kernels and 10 FM complex

inputs. Could less FM inputs be used? In a further experiment the number of RI kernels was fixed and the number of complex inputs was varied. The results are given in Figure 6–12. A classification rate of 76.5% was achieved with only 7 inputs; over-fitting was occurring with 10 inputs.

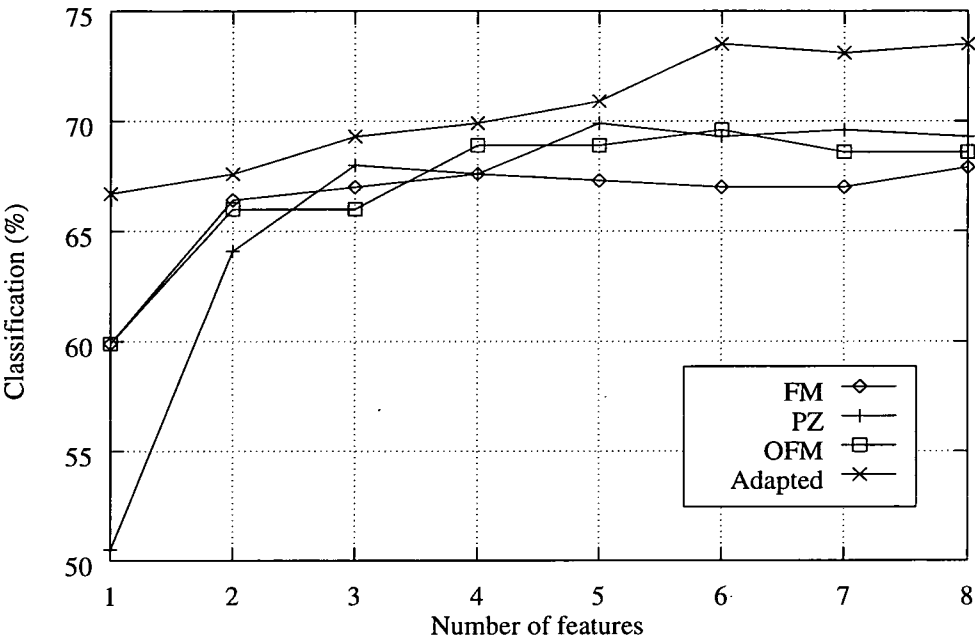


Figure 6–11: Seascape: Increasing the number of feature kernels.

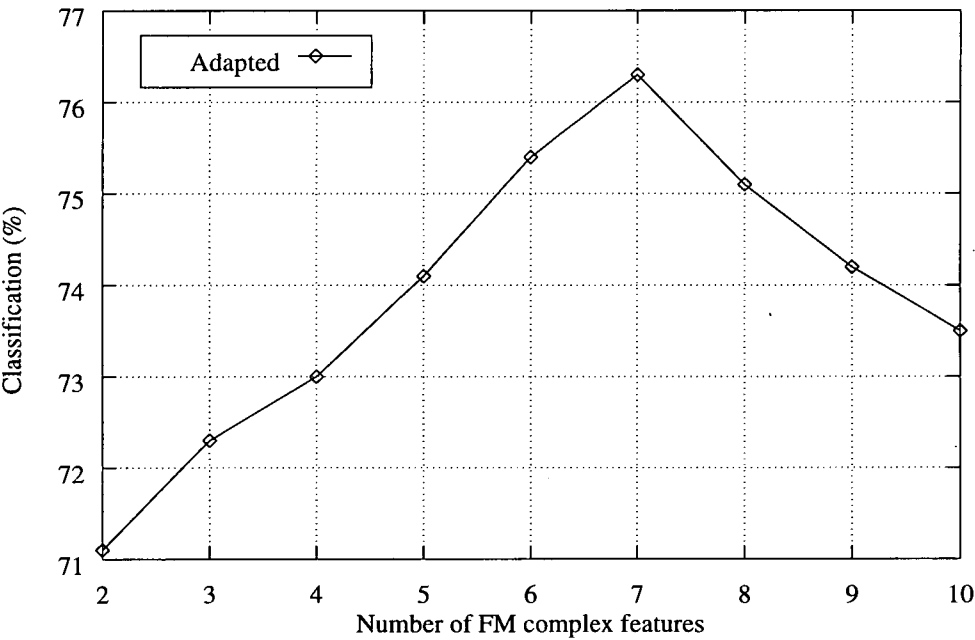


Figure 6–12: Seascape: Increasing the number of complex FMM inputs.

However, there were several outstanding questions. How would these new features cope with additive image noise, how easy could the FMM features of different  $m$  be incorporated into the model, and what polynomials were being generated in the adapted model? The first and third questions are answered in Figures 6–13 and 6–14.

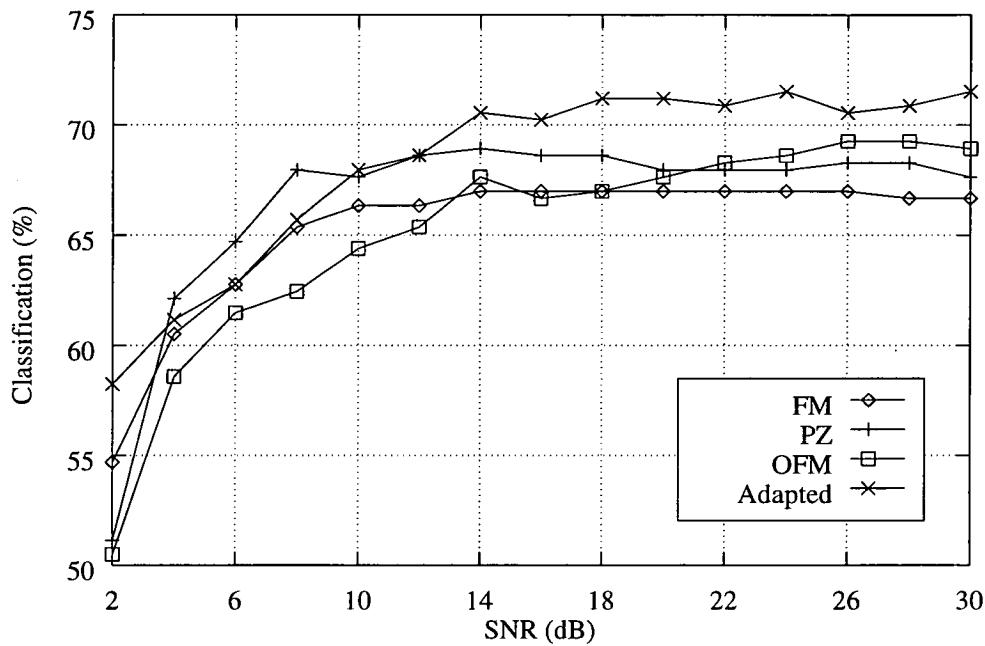


Figure 6–13: Seascape: Noise results for the adaptive model.

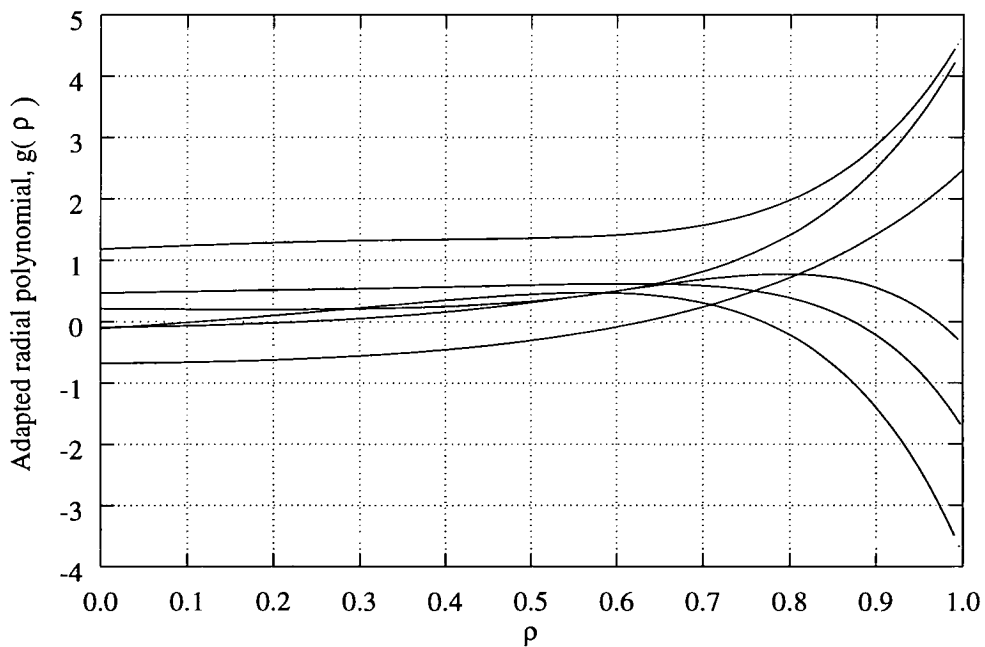


Figure 6–14: Seascape: Final radial polynomials ( $m = 2$ ) for the adaptive model.



As can be seen the classifier coped as well as the other moment-based features. The surprising result were the radial polynomials. These were order 6 polynomials ( 7 FM complex inputs ) extracted from an optimised 6 RI feature kernel network. The extremities were very important for discrimination. The confusion matrices indicated that the sailboats had separated from the other two classes successfully. The motor boats and the buoys were the source of confusion in this case.

These results have shown that improved RI classification was achieved using the adaptive technique. However, only one particular circular harmonic order was examined. What would happen when FMM features, with different  $m$ , were used? The next experiment used the identical model as before but with  $m = 4$  inputs included. This network was significantly harder to optimise. The resulting  $m = 4$  polynomials were very different, as shown in Figure 6–15. This figure showed that significant alterations had occurred during optimisation and that more emphasis had been placed by the polynomials around the object centre. At the higher frequency the centre was more attractive as a source of class discrimination.

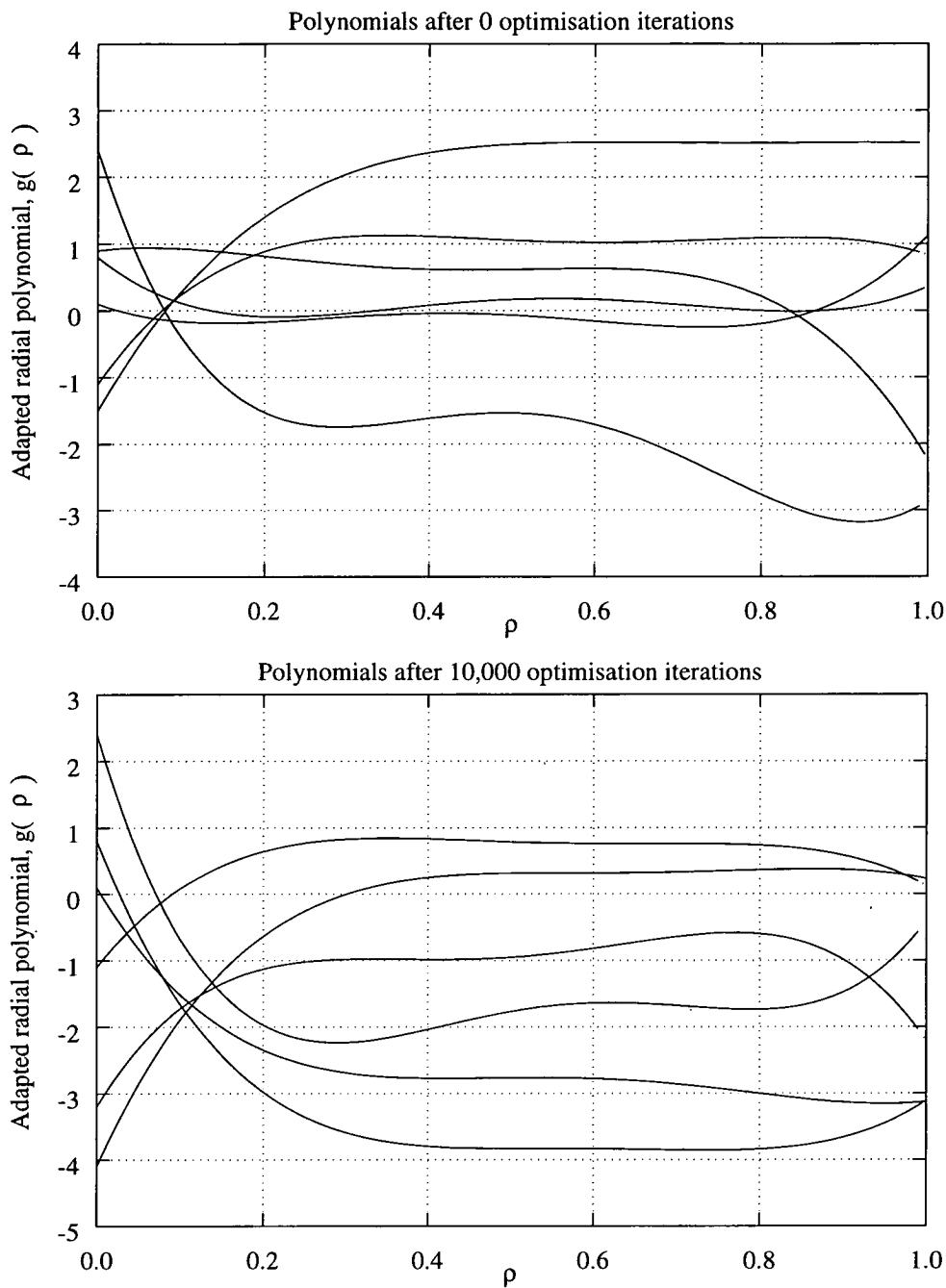
Generating a model that would include even more  $m$ , and thus become useful, would have required large numbers of inputs and connections. The error surface was also known to be complicated, and even the simplex method on occasion failed. Thus, other results have not been included in this thesis. A more simplistic adaptive RI feature classifier was sought for the project.

## 6.9.2 RI through $\theta$ normalisation

Another approach was to transform the image such that RI was naturally incorporated into the new image:  $\theta$ -normalisation. As has been seen, this process normalises for object rotations using a linear shift in the  $\theta$  direction equal to the circular mean,  $\bar{\theta}$ , determined by solving  $\cos \bar{\theta} = C(\theta)/R(\theta)$  or  $\sin \bar{\theta} = S(\theta)/R(\theta)$  where  $R(\theta) = (C^2(\theta) + S^2(\theta))^{1/2}$  and

$$C(\theta) = \int_0^1 \int_0^{2\pi} \cos \theta f(\rho, \theta) d\theta \rho d\rho \quad \text{and} \quad S(\theta) = \int_0^1 \int_0^{2\pi} \sin \theta f(\rho, \theta) d\theta \rho d\rho. \quad (6.14)$$

The new image,  $f(\rho, \theta + \bar{\theta})$ , is then invariant to the initial rotation of the image. This approach allowed for the direct application of the standard adaptive feature extraction techniques discussed in Chapter 5. This was a major advantage.



**Figure 6–15:** Seascape: 6 radial polynomials ( $m = 4$ ) after 0 and 10,000 iterations.

The  $\theta$  normalisation process was applied to the seascape objects which had been mapped to a 20x72 polar coordinate system. The data was then split, as described in previous chapters, into three individual sets and two sets of experiments were performed. The first used a linear adaptive network, and the second an extra nonlinear layer, as in Chapter 5.

6.9.3 Linear adaptive kernel

The initial adaptive experiment adapted two positional kernel parameters ( $x_0, y_0$ ) of a simple Gaussian kernel, with  $a = 2.50$  and  $b = 1.25$ . The positional parameters, which were found to be some useful in the Cartesian experiments of Chapter 5, were optimised as before. The next test allowed all four parameters of the Gaussian parameters ( $a, b, x_0, y_0$ ) to be adapted in the hope of improving classification. Finally, using the real part of the Gabor kernel, six parameters were used. In each test the number of kernels,  $N$ , was varied and the optimisation process applied for 1000 iterations. Results are given in Figure 6–16 where each point represents the mean value, over 10 different random splits of the data, with a standard deviation of approximately 0.5%. The results indicated that for large values of  $N$ , with the seascape data, simple Gaussian kernels, with four adaptive parameters, were sufficient to perform the classification task. The final classification rate of 78.0% was 2.0% greater than the FM linear classifier results given in Table 6–2.

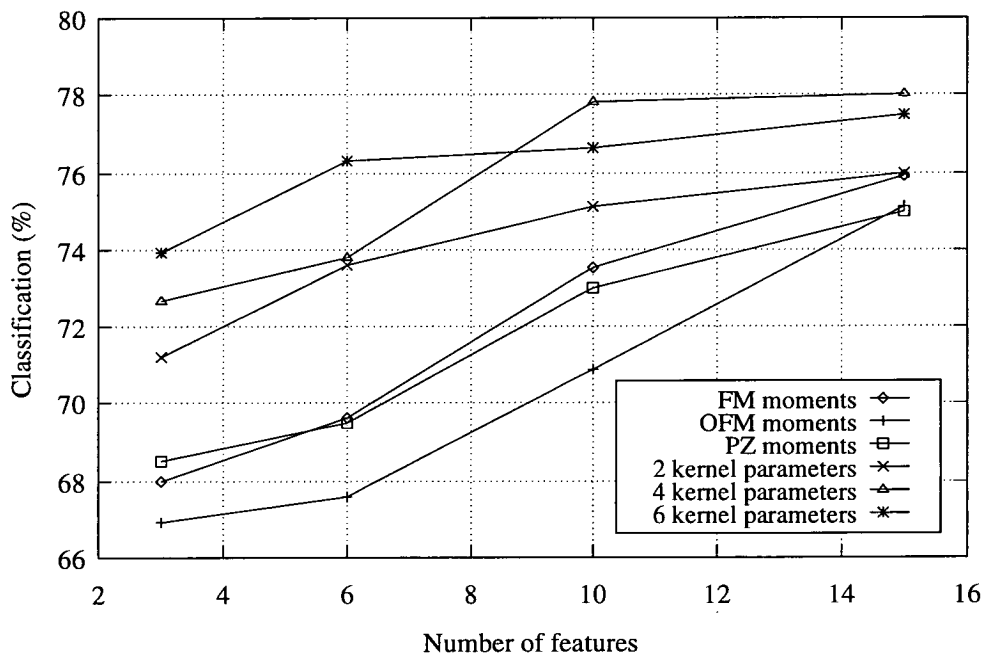


Figure 6–16: Seascape: Adaptive RI linear classification results.

One of the most noted attribute of moment-based features is their insensitivity to image noise. Pseudo-Zernike moments have been found to be less affected by noise than, for example, Fourier-Mellin or Zernike [124]. Figure 6–17 shows how the various adaptive linear models

were affected by additive Gaussian noise in comparison with the PZ moments. These models were more sensitive to noise than their moment-based counterparts.

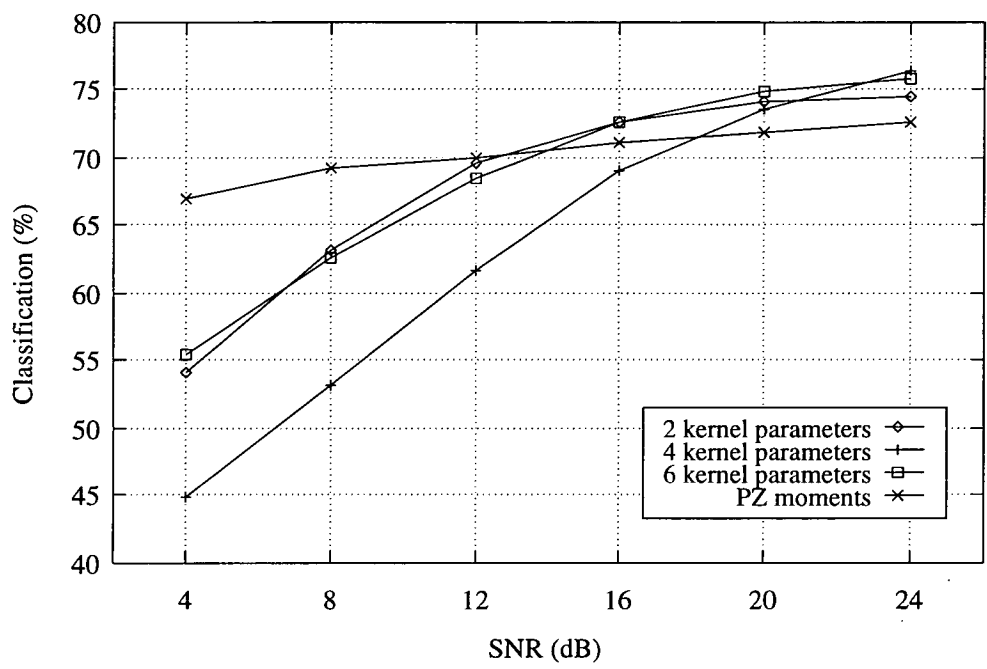
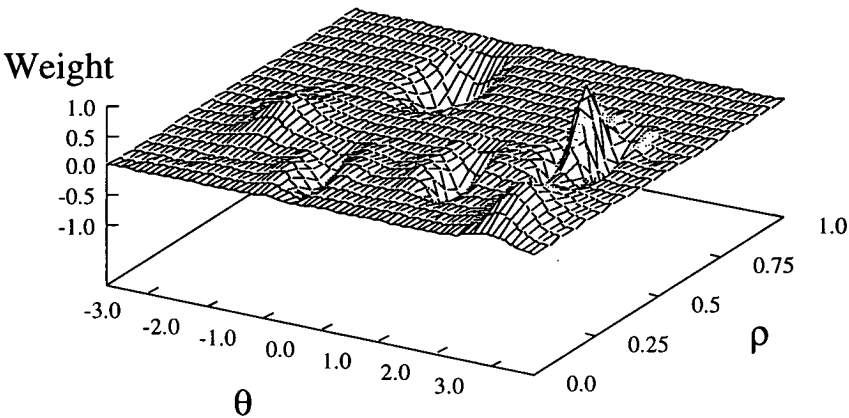


Figure 6–17: Seascape: Effect of noise adaptive RI linear model performance.

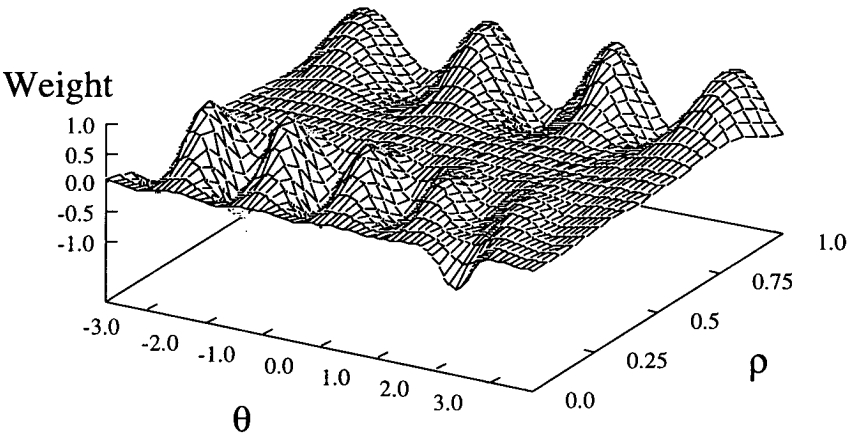
Figure 6–18 shows how the effective kernel linear classifier (i.e. the weighted sum of kernels) for the sailboat class changed during optimisation. The classifier had more, smaller, kernels around the centre ( $\rho = 0$ ) and used fewer, broader kernels towards the extremities of the image. These areas included the tops of the sailboat masts only. The motor boats and buoys, with more symmetric pixel distributions, had consequently more energy near the extremities. This tied in exactly with the results that were discovered with the adaptive complex kernel classifier. There was no kernel influence in the mid region.

6.9.4 Nonlinear adaptive kernel

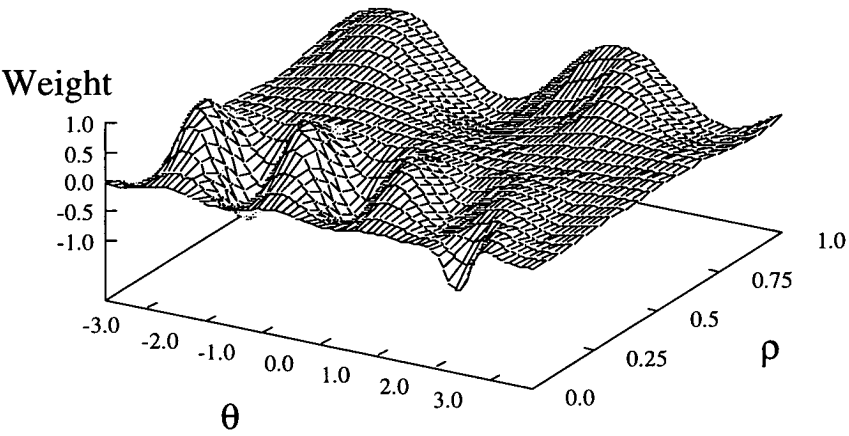
As in Chapter 5 the next step was to include the usual MLP sigmoid layer of processing elements to provide nonlinear classification of the RI images. The discussion of this network was covered in the previous chapter, so it suffices to simply provide results for this new RI image database. The only difference is the image sampling.



(a) 0 iterations, 43.0%



(b) 500 iterations, 70.8%



(c) 1000 iterations, 78.0%

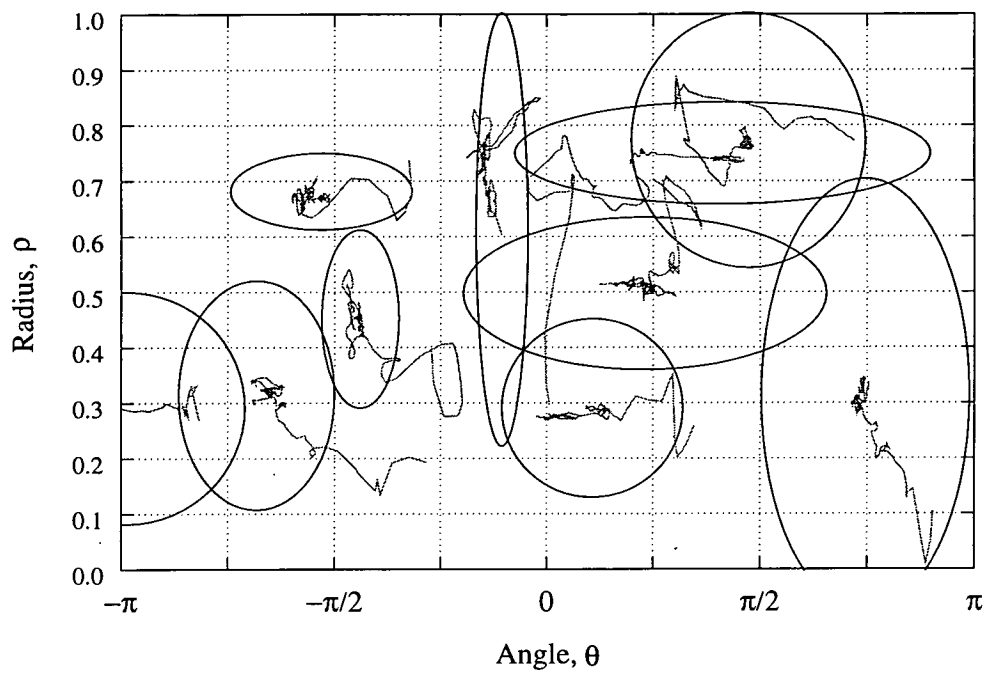
**Figure 6–18:** Seascape: Combined filter weights for sailboat class.

Using the RI image database used in the last section several nonlinear networks were optimised and tested. The classification performances for networks of varying flexibility is recorded in Table 6–4. The 4-parameter  $(x_0, y_0, a, b)$  Gaussian kernel, that worked well in the non-RI case, was also used here and a limit on the number of parameters was set to 180 ( similar to a standard 8 hidden node MLP with 15, for example, moment features.)

Kernels	Number of hidden units					
	2	4	6	8	10	12
3	73.25	76.0	76.5	77.25	78.0	77.75
6	74.5	80.5	80.5	80.5	78.75	-
10	76.75	82.25	83.25	85.5	-	-
15	79.0	82.5	82.75	-	-	-
15 fixed PZ moment features:						
	70.25	75.5	80.5	81.75	-	-

**Table 6–4:** Seascape: RI features with a rotated database. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests.

This table shows that with similar numbers of parameters the adaptive technique matched, and surpassed, the performance of the classification achieved with the best fixed RI features. The maximum classification rate achieved was 85.5% using 10 kernels and 8 hidden nonlinear hidden units. Figure 6–19 shows how this was achieved in terms of the final centroid and shape of the feature extracting kernels. The ellipses denote equal kernel values, and the grey lines denote the kernel trajectories during optimisation. Again, the kernels tended towards the centre of the image (small  $\rho$ ) and concentrated less on the extremities.



**Figure 6-19:** Seascape: Tracking of kernel centroids during optimisation and final shape.

6.10 Review

This chapter has reviewed invariant object recognition. This is a subject of great importance in real ATR systems. One particular type of invariance, planar rotation, was discussed in detail and two methods for incorporating rotation invariance into the adaptive feature extraction classification, described in the last chapter, were outlined. The first, based on adaptive complex kernels similar to moment features, were shown to be effective but very difficult to optimise. A second method, based on preprocessing the images such that RI was already incorporated into the data, proved much easier to use as it was a simple extension of the previous adaptive work. Results were given for both methods using the *real* IR seascape database containing ideally segmented objects. It was now time to examine the effects when objects were generated by a non-ideal segmentation process and how that would effect classification performance.

---

## Chapter 7

# Integration into the ATR environment

---

In the previous chapters several adaptive feature extraction and classification models were reported as attractive solutions to the invariant classification of real, well-segmented, IR objects. These models required small numbers of model parameters to be estimated, yet still allowed for model estimation with respect to the original image data. However, these results were based on one particular ATR environment using a system that was assumed to generate ideally segmented objects. The next step in the project was to examine the effects on the adaptive classifiers when these assumptions were no longer valid, what could be done to minimise performance degradation, and finally, what would happen when other ATR situations were considered. Three particular issues had to be addressed:

- The effect on classification due to rogue data produced by a non-ideal segmentation algorithm. This included
  - I : The influence of clutter, everything else that the segmentation module wrongly assumed was a target.
  - II : Phenomenological effects due to object occlusion as a result of a cluttered scene.
  - III : The effect of segmentation failure on classification.
- The identification and differentiation of, and between, clutter and poorly-segmented objects.
- The adaptability of the system to a new environment.

Of course, there were other potential hazards but, within the scope of the project, object clutter, occlusion, and poor segmentation quality were found to affect most severely classification performance. This chapter considers each of these problems in turn, and how each



affected generalisation. The chapter begins by reporting the effect of object occlusion had on the seascape object classifications and follows with a discussion of the techniques that were employed to differentiate between the clutter and the poorly-segmented objects.

In the requirements, set down in Chapter 1, it was stated that the classification system *should be easily adaptable to a new environment, and target type, by an unskilled or automated, operator*. The classifiers discussed, in theory, should be able to perform this task, and this chapter finishes by reporting on the results of the adaptive models with a completely new, real, infrared database.

Before discussing the application of the new models to the BASE data a few important points need to be highlighted. These models will be used in many different situations, and this was the reason for a requirement that the system should be readily adaptable. It is suggested that the problems with BASE data are common with many other real-world situations in that:

- The vast majority of real-world environments will have artifacts with similar properties to those the system is interested in classifying. In this way clutter generation is inevitable.
- At the time of writing, and for the foreseeable future, perfect automated object detection and segmentation processes do not exist. This means that object extraction will fail at some point.
- There will be processes at work, such as occlusion, which will affect object representation.

The problems with the BASE data is simply a subset of a more generic problem of object generation. The difference being that each new application will have a different distribution and probability of occurrence for the rogue data.

## 7.1 The effects of rogue data

Throughout this thesis various assumptions have been considered concerning the validity of the results of a new ATR classification module. For example, the work in Chapters 4, 5, and 6 assumed that the objects provided were of ideal segmentation quality. The results in Chapter 3 showed that this was not being achieved with the original segmentation algorithm and even though there is no agreement in the literature over what constitutes an ideal object segmentation<sup>1</sup>, changes in segmentation quality would be highly likely to affect classification performance. With a more advanced segmentation algorithm the likelihood of a more predictable object extraction would be greater, though not guaranteed. For example, in the seascape database the clutter had properties similar to the objects of interest. Consequently, the effects of clutter and poor object segmentation on classification had to be considered. The current segmentation algorithm was thus useful as it provided large numbers, of what were termed, *rogue data* to test the effects of segmentation failure in the extreme.

### 7.1.1 Clutter

Clutter was introduced in Chapter 3 as the artifacts extracted by the segmentation process that were, in fact, of no interest. They were non-objects. They were extracted because they had similar properties to all the other types of objects, for example, they radiated heat. So unless one particular property was available to distinguish the objects, clutter would always be produced.

The classifiers that had been designed at this stage had no knowledge of clutter, and no rejection method. The clutter was classified according to the position in feature space in which they occurred, and the decision boundaries created using the object data. Using two different types of feature the 956 cases of seascape clutter were classified, using both a linear and 7-NN classifier developed on 3-category, non-RI (Chapters 4 and 5) and RI (Chapter 6), well-segmented data. The results are given in Table 7-1.

---

<sup>1</sup>Ideal segmentation in this thesis has been assumed to be that of skilled hand segmentation by a human, though other segmentations may have led to easier classification, for example, by not segmenting sails, only masts.

Feature	#	RI	Classified as			Total
			Sail	Motor	Buoy	
Linear:						
Gaussian	16	No	85	<b>637</b>	234	956
Zernike	15	Yes	295	<b>485</b>	176	956
7-NN:						
Gaussian	16	No	137	<b>704</b>	115	956
Zernike	15	Yes	<b>587</b>	284	85	956

**Table 7–1:** Seascape: Clutter classification using a 3-category linear and 7-NN classifier.

In the non-RI case the clutter was being classified as motor boats. Clutter did, in fact, tend to be thin and horizontal, such as the wash from boats and sections of coastline. However, in the RI situation this horizontal information was lost and clutter was classified either as sailboats or motor boats, dependent on the type of feature. This inability to reject data would lead to a high false alarm rate on particular classes of object. This was not satisfactory, especially if the class was to be of particular importance.

7.1.2 Occlusion

Another potential source of danger with the seascape data occurred because of the highly cluttered environment where objects, and objects and clutter, such as rocks or thick smoke, would overlap in the two-dimensional image representation. In the previous chapters the adaptive models were tested for their ability to classify in the presence of noise. This is a standard test applied to image-based classification problems but not necessarily, given the quality of modern day sensors, a very realistic problem. Occlusion though was a serious problem with the seascape data as it caused segmentation failure; either entities were combined into single objects, or only sections of an object were extracted. Occlusion was thus labelled as a special case of segmentation failure. The difference with the ordinary single object segmentation failure was that no amount of corrective segmentation could, without complex extrapolation, derive the true object segmentation.

### 7.1.3 Segmentation failure

In Chapter 3 individual object segmentation failure during the creation of the seascape database was discussed. Different severities of failure were introduced and catalogued. It was now appropriate to see how classification degraded with segmentation quality. Thus, a classifier was trained using purely well-segmented data and tested separately using:

- 1609 objects with good internal, and external quality (EX0 IN0).
- 385 object internals that were slightly either too large or small (EX0 IN1-2).
- 568 objects with complete internal segmentation failure (EX0 IN3)
- 466 objects with external segmentation failure, regardless of internal failure (EX1-3).

The results for various types of features using a 7-NN classifier are given in Figure 7-1. For small inaccuracies in the segmentation (EX0 IN1-2) there was, for all classes, a slight degradation in performance, ranging between 1 and 20%. The motor boat and buoy classes typically suffering the worst. However, when the internal segmentation failed completely (EX0 IN3) large differences occurred, dependent on feature type and class. With every feature the buoy class suffered very badly. In Chapter 3 it was noted that the main differences between the buoys and the sailboats was the grey level distribution. They tended to have very similar outlines. Thus, when the buoy internal segmentation failed, all that was left was the outline, and the object was classified as a sailboat. This was confirmed on examination of the confusion matrices. The motor boats, in the RI cases, also suffered for similar reasons. In the non-RI cases the motor boats exhibited different behaviour for different features. In the non-RI case with Gaussian features, classification was performed using the fact that there were object sections to the extreme right and left (the bow and stern) but little in the top and bottom thirds. Gaussian features, and subsequently, classification were thus unaffected for this class if significant central portions were missing. With Fourier features losing large amounts of data caused significant changes in the frequency content of the image due to the thin, high frequency, skeletal structure of the (EX0 IN3) objects. This caused large changes in features.

The sailboat classification results were the most interesting. There appeared no degradation in classification with segmentation failure. In fact, in some cases there was an improvement

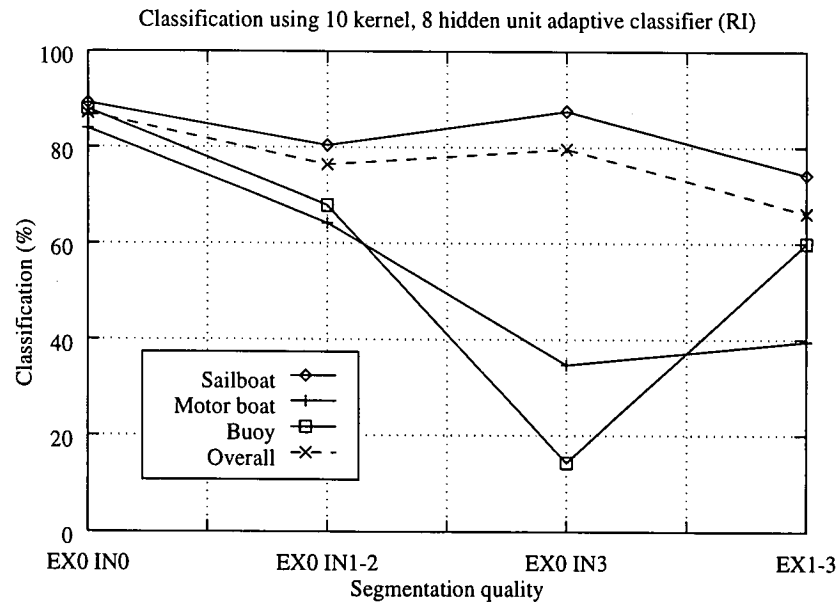
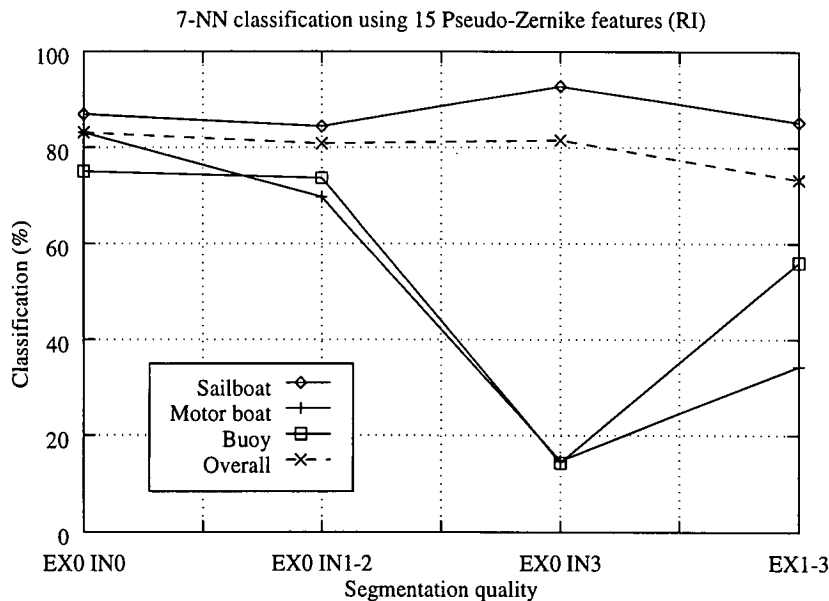
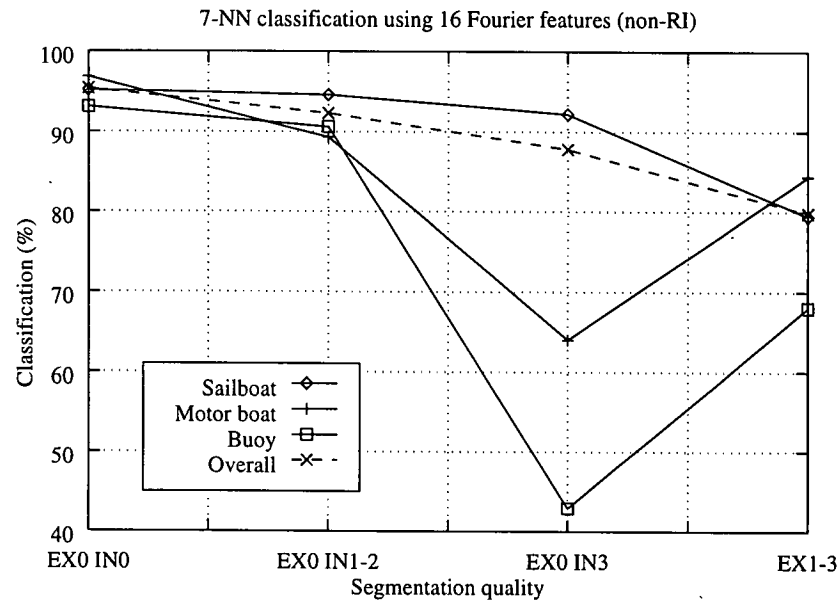
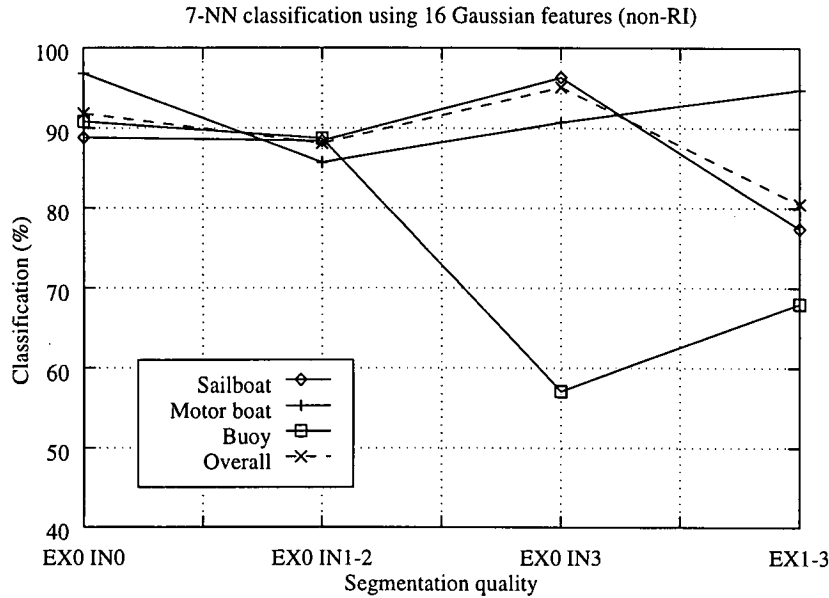
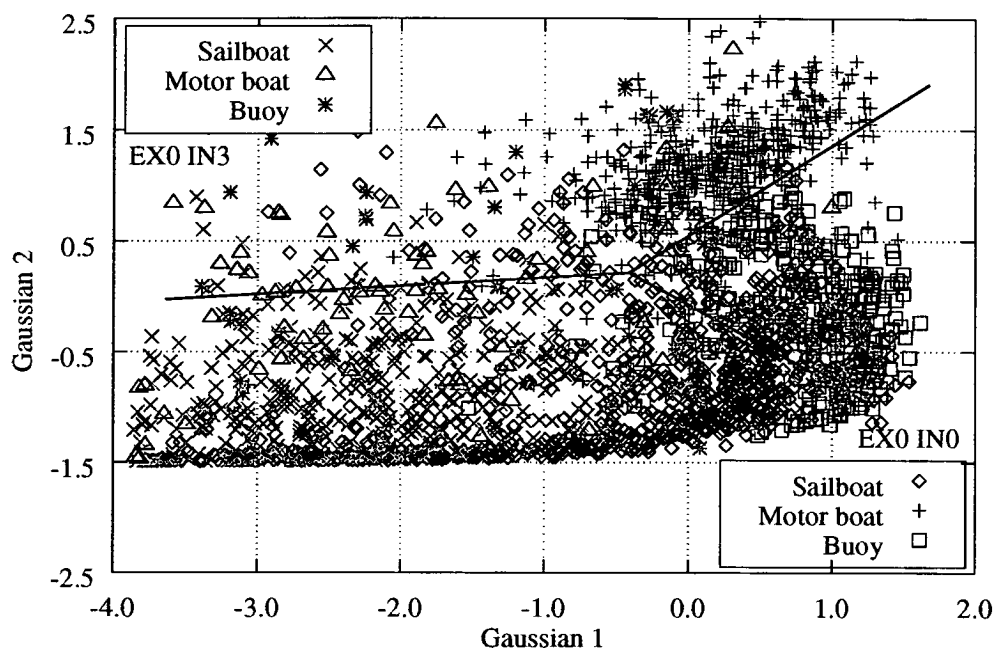


Figure 7-1: Seascape: Classification versus segmentation quality.

in classification. The features were found to have changed drastically in value with change in segmentation quality. However, these new features had not transgressed any decision boundary, unlike the other two classes, and were in fact further away from the decision boundary, and subsequently much less likely to be misclassified. This is demonstrated in Figure 7–2 using two Gaussian features and a linear decision boundary created using the well-segmented data.



**Figure 7–2:** Seascape: Badly segmented Gaussian feature distribution.

The overall classification remained relatively high due to the fact that the majority of poorly-segmented objects were sailboats.

It was also found that if the poorly-segmented objects had been included in the original database used in Chapters 4-6 they would have introduced, not only degraded performance in the test data, but also significantly affected the decision boundaries created. Remember that these poorly-segmented objects, away from the other well-segmented data points, would have generated large sum-of-squares errors, and would have dominated the final decision boundaries. This is demonstrated in Table 7–2 where two 8 hidden node MLP’s were trained using either well-segmented data, or a mixture of both well and poorly-segmented object features. The resulting networks were both tested using similar types of databases, but derived from different objects. The entries in the tables are the resulting classification performances in the four

different cases, with the usual one standard deviation error in brackets. Sixteen fixed Gaussian features were used.

The results for the networks optimised with well-segmented data, and tested with well-segmented data, were taken from Chapter 4. When these networks were tested with the mixture of both well and poorly-segmented objects the classification performance dropped off significantly. This was due to the existence of the motor boats, and buoys, across the decision boundaries, as shown earlier in Figure 7–2.

Test set	Training set	
	Well-segmented only	Mixture of both
Well-segmented only	92.75% (1.0)	91.5% (0.9)
Mixture of both	86.25% (1.2)	89.75% (1.0)

**Table 7–2:** Seascape: Effect of segmentation quality on MLP training and classification. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests.

When optimised with all the non-clutter objects, the network performance using well-segmented test data was slightly degraded due to the warping of the decision boundaries by the rogue data points. However, in this case the drop in classification was far less noticeable with the mixed data test set as the boundary formed had adjusted for the rogue data, especially the sailboats. So optimising the classifiers with all the non-clutter data reduced the effect of bad segmentation but there was still no way of identifying the rogue points.

## 7.2 The identification of rogue data

Of course, the first question is why there is a need to identify rogue data? In the previous section it was seen that by including poorly-segmented data in the training set performance degradation could be reduced. However, if a test object could be identified as being poorly-segmented, action could be taken by, for example, re-segmenting the object with different, possibly more suitable, segmentation parameters and classifying again. Furthermore, if the rogue data was clutter, and did not belong any object class then a mistake would always occur. So, was there any way of examining the outputs of the classification stage to identify both these types of rogue data?

### 7.2.1 Classifier outputs and *a posteriori* probabilities

As discussed in Chapter 2 most classification tasks operate by allocating an unknown feature vector to one of  $C$  pre-defined classes,  $\omega_i$ , such that the *a posteriori* probability,  $P(\omega_i | \mathbf{x})$ , is maximum. Clutter is not a pre-defined class, it has a separate distribution, and thus  $\max P(\omega_i | \mathbf{x})$  is nonsensical. Rogue data, in terms of poorly-segmented objects, however, is dependent on the class definition. If the class is defined as including only well-segmented examples then this rogue data, also has a separate distribution. Again  $\max P(\omega_i | \mathbf{x})$  is irrelevant. If rogue data is included within the class it will be reflected in the as either another mode, or extended tail, in the class distribution.

Figure 7–3 demonstrates the effect a poorly-segmented object had on the output of a classifier designed using a mixture of both well- and poorly-segmented data. The object existed in the tail of the sailboat distribution. This time  $P(\omega_i | \mathbf{x})$  predicted the object class correctly, but with a magnitude greater than that of the well-segmented data. However, as seen in Figure 7–2 this will not always be the case and  $P(\omega_i | \mathbf{x})$  could have easily been lower for the poorly-segmented cases of class  $\omega_i$ . So  $P(\omega_i | \mathbf{x})$  could not be used to identify this type of rogue data. The rogue data had to be considered as separate from pre-defined, non-clutter, well-segmented object classes.



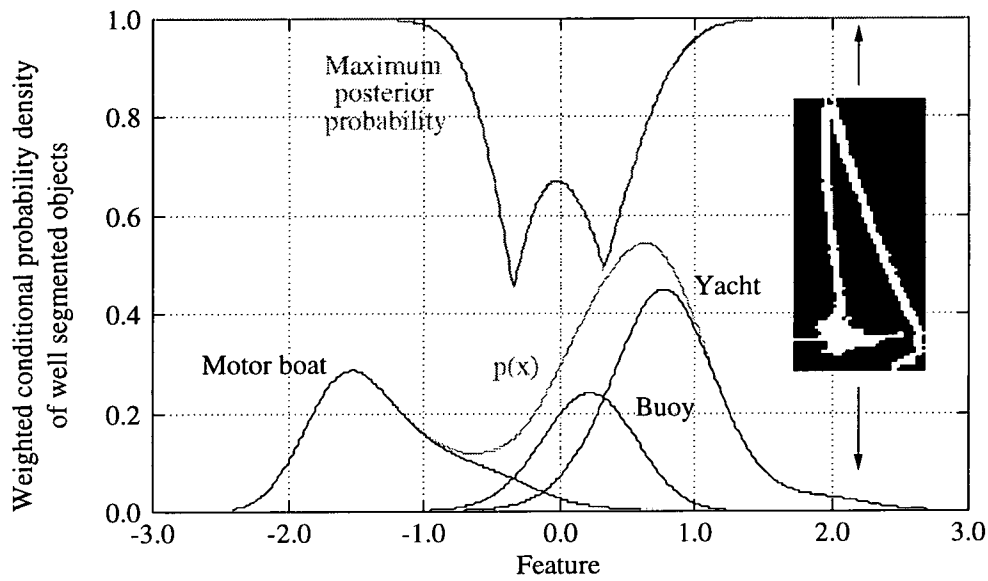


Figure 7-3: Seascape: Poorly-segmented object classification.

Two methods were examined as possibilities for identifying rogue data. The first included the rogue data as an extra class providing  $C + 1$  class discrimination. This is the usual method for categorising clutter and has been employed by BASE [49]. The second method assigned a measure of how alike a pattern was to anything in the  $C$ -class training set, with highly novel patterns being labelled as possible rogue data.

7.2.2  $C + 1$  classification

This approach of using an extra class in the discrimination procedure assumed that the rogue data had a distribution that was adequately represented by the sampled data. If this assumption was correct then classification could be applied, as before, with good generalisation capabilities.

Initially, each of the two types of rogue data were examined separately. Clutter was first considered to have a broad distribution that covered the entire feature space. However, this was found to be correct as the clutter features were products, of not just the segmentation, but also the feature extraction process. If clutter were localised, as was suggested for the seascape data by examination of various feature spaces, then treating clutter as an extra class was an appropriate proposition.

Using both features from Chapter 4 and the adaptive feature extraction networks from Chapters 5 and 6 a four class classification table for the seascape data was created. There were 956 samples in the clutter class. The results are given in Table 7–3. The adaptive model results are given for both the linear and nonlinear approaches. For the linear adaptive model nine 4 parameter ( $x_0, y_0, a, b$ ) Gaussian kernels were used. For the nonlinear model only six of these kernels were used, combined with four nonlinear nodes.

Feature	#	Linear	7-NN	MLP
Fixed non-RI features:				
Gaussian	16	73.25 (1.6)	81.25 (2.4)	84.5 (1.2)
Legendre	15	72.5 (2.5)	81.5 (2.5)	81.0 (1.6)
Fourier	16	76.5 (2.2)	85.75 (1.7)	84.0 (1.2)
Fixed RI features:				
OFM	15	60.0 (2.9)	70.5 (1.5)	73.5 (2.1)
Pseudo-Zernike	15	64.0 (1.9)	75.25 (1.5)	74.25 (2.2)
Adaptive non-RI model:				
		76.5 (1.4)		84.75 (1.5)
Adaptive RI model:				
		67.25 (1.5)		74.25 (1.0)

**Table 7–3:** Seascape: Clutter classification using a 4-categories of data. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests.

The additional clutter class produced an approximately 10% decrease in classification performance when using the same fixed features as before. The fixed Fourier features, again, provided for superior classification whilst the adaptive networks, as can be seen, have compensated for the new class, scoring as high and higher, than the best fixed feature results. The confusion matrices for incidents of the non-RI and RI adaptive linear classifications are provided in Table 7–4. In the non-RI case, as previously suspected, the main confusions were between the clutter and the motor boats, as well as, the usual buoy-sailboat confusion. With

the RI adaptive model, again as suspected, the confusions were much more spread out across the classes with objects being confused for clutter equally across all the non-clutter classes.

Guess	Correct class			
	Sail	Motor	Buoy	Clutter
Sail	104	1	<b>26</b>	10
Motor	0	79	3	<b>19</b>
Buoy	2	2	28	9
Clutter	3	<b>14</b>	0	100

Guess	Correct class			
	Sail	Motor	Buoy	Clutter
Sail	96	3	11	<b>20</b>
Motor	5	65	3	<b>25</b>
Buoy	<b>16</b>	<b>28</b>	43	<b>22</b>
Clutter	2	0	0	71

(a) non-RI adaptive model (77.75% correct)

(b) RI adaptive model (66.25% correct)

**Table 7–4:** Seascape: Confusion matrices for the linear adaptive networks with a clutter class.

So, overall, the addition of the clutter class produced a reduction in the classification performance, and this was due to the similarities that existed between the clutter and certain object features. But what about the rogue data generated by poor object segmentation? There were 1419 examples of poorly-segmented objects in the seascape database. How would these classify? The results using this data as a fourth class, instead of the clutter, are given in Table 7–5.

Feature	#	Linear	7-NN	MLP
Fixed non-RI features:				
Gaussian	16	62.0 (1.3)	74.5 (1.4)	75.0 (0.9)
Legendre	15	59.75 (1.7)	75.0 (1.7)	75.0 (1.2)
Fourier	16	63.75 (0.8)	77.0 (0.7)	77.75 (1.1)
Fixed RI features:				
OFM	15	58.5 (1.2)	71.75 (0.9)	71.75 (1.2)
Pseudo-Zernike	15	59.5 (1.0)	73.75 (1.1)	74.0 (1.5)
Adaptive non-RI model:				
		64.0 (1.3)		78.25 (1.5)
Adaptive RI model:				
		60.0 (1.1)		75.0 (1.3)

**Table 7–5:** Seascape: All badly segmented data classified using 4-categories. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests.

The results indicated that there was significant overlap between the object classes and the poorly-segmented rogue data. This was expected as this rogue data also included objects which were only slightly mis-segmented, (EX0 IN1-2) for example. For the re-segmentation project that was run in parallel with this project it was useful to examine whether the objects with gross segmentation defects could be identified.

When the extremely poorly-segmented objects were used class separation was improved. This is shown in Table 7–6 where 568 (EX0 IN3) data samples form the fourth class. None of the adaptive models were applied to this data due to project time restrictions.

Feature	#	Linear	7-NN	MLP
Fixed non-RI features:				
Gaussian	16	78.25 (1.8)	85.0 (1.5)	85.0 (1.3)
Legendre	15	77.0 (1.4)	85.5 (1.4)	85.5 (1.2)
Fourier	16	79.25 (1.5)	89.0 (0.8)	88.0 (1.5)
Fixed RI features:				
OFM	15	70.25 (1.5)	81.5 (1.2)	82.0 (1.0)
Pseudo-Zernike	15	74.75 (1.6)	84.0 (1.1)	84.0 (0.9)

**Table 7–6:** Seascape: (EX0 IN3) badly segmented data classified using 4-categories. Each score is the mean percentage classification over 10 different samples each consisting of 400 test vectors. The value in parentheses is the standard deviation over the 10 tests.

The classification rate improved significantly which indicated that it was possible to identify this type of rogue data. Table 7–7 provides confusion matrices for two of the fixed feature classifications using a 7-NN classifier. It was unsurprising to find that the sailboats were the main source of confusion and they were most prone to drastic segmentation failure with their skeletal representations leaving little left to provide useful classification information.

Guess	Correct class					Guess	Correct class				
	Sail	Motor	Buoy	(EX3			Sail	Motor	Buoy	(EX3	
				IN0)						IN0)	
Sail	111	1	0	12		Sail	103	15	4	6	
Motor	0	105	1	4		Motor	9	88	7	6	
Buoy	11	0	71	5		Buoy	9	3	61	1	
(EX3	8	0	0	71		(EX3	9	0	0	79	
IN0)						IN0)					

(a) Fourier features (89.5% correct)

(b) Zernike features (82.75% correct)

**Table 7–7:** Seascape: Confusion matrices for the fixed feature 7-NN classifiers with an (EX3 IN0) class.

The  $C + 1$  class algorithm was shown to be successful in identifying both clutter, and very badly segmented object, rogue data points. The method allowed ease-of-use of the adaptive feature extraction techniques described in the previous chapters. However, the method made an assumption concerning the distribution of the fourth rogue class. The next method provided a solution in the situation where that assumption was not valid.

### 7.2.3 Novelty classification

In the previous section the rogue data was identified by treating this data as another object in the classification system. This was made possible by the assumption that the detection and segmentation process will generate rogue data drawn from a fixed distribution. Thus, given enough examples of the rogue data from the object generating process, *which are representative of the fixed rogue data distribution*, classification, and subsequently rogue data identification, can be performed. With the seascape data this was shown to be a successful approach and allowed the adaptive feature extraction models to be incorporated.

The second rogue data identification method, described in this section, can be used when the rogue data distribution is unknown, severely undersampled or not constant. In these conditions, which would occur when an automated, adaptive, segmentation process was considered, the  $C + 1$  performance would be severely degraded.

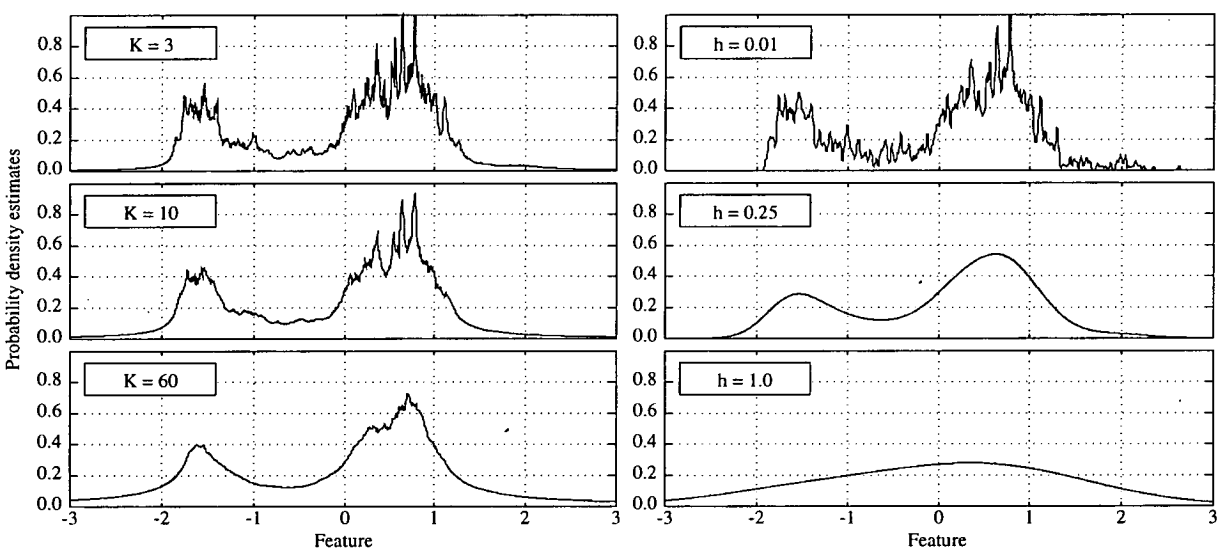
This secondary approach makes use of the fact that the ideally segmented, non-clutter, object feature distributions will, by definition, remain constant if the environment and segmentation algorithms are changed. Identification of the rogue data can be performed by rejecting data that falls out of the region of the object feature distributions by some pre-set threshold.

This secondary method would require a measure of the degree of novelty of a test pattern, how much the feature was away from the norm. Figure 7-3 indicated that the inverse of the unconditional probability estimate,  $\hat{p}(\text{non} - \text{rogue}x)$ , was a prime candidate. Indeed,  $\hat{p}(x)$  had been used in other fields as a measure of novelty where low values of  $\hat{p}(x)$  indicated high novelty [14].

Bishop suggests classifying objects as novel if the estimated unconditional probability of a given feature falls below a given threshold value [14]. The threshold value that separates

the non-novel from the novel, rogue, data class may be determined using the Bayes rule. The probability density  $p(\text{rogue})$  is assumed to be uniform<sup>2</sup> over volume the feature space that could be possibly be covered. Practically, this volume is quite arbitrary and the threshold has to be determined by experimentation. For the project, it was not necessary to be very accurate with the threshold, provided it erred on the side of identifying more novel data than expected, as this data was to be passed to a secondary assessment module. An inaccurately positioned threshold would mean simply more work for this stage. The question remained how to estimate  $p(\text{non} - \text{rogue})$ .

There exist several methods for estimating probability densities including kernel density estimators such as the Parzen window approach,  $k$ -NN and Gaussian mixture models [103, 134]. Each of these methods require a set of smoothing parameters. The estimates for one particular seascape Fourier feature, on the right in Figure 7–4, were derived using a Gaussian kernel estimator with various Gaussian widths,  $h$ , of 0.01, 0.25, and 1.0. Visually with



**Figure 7–4:** Examples of two type of density estimator.

$h = 0.25$  the multi-modal structure of the distribution could be seen but without the unwanted sharp peaks. Unfortunately, determining smoothing parameters, even using advanced cross validation techniques, in multi-dimensions is notoriously difficult [97]. The  $k$ -NN estimator was a more intuitive technique requiring adjustment of a single parameter,  $k$ . It was also

<sup>2</sup>Though this is a very unsafe assumption as the rogue data is the result of a deterministic segmentation process.

simple to implement, had been encountered already with classification, and so was used in the project for the novelty tests. The estimator was determined as described in Chapter 2. As stated in Chapter 2 the  $k$ -NN estimator is not strictly speaking a true density estimator but was effective enough for the purpose of the project. The effect of different  $k$  is shown on the left in Figure 7-4.

In practice, this form of thresholded  $\hat{p}(\mathbf{x})$  classification using  $k$ -NN density estimation equates to thresholding on an Euclidean distance in feature space. In fact, this type of novelty classification is very similar to the  $C + 1$  algorithm, described in the previous section, but only if the rogue data has a wide, uniform, distribution, is fully representative for determining classification boundaries and these boundaries suitably flexible.

The  $k$ -NN based novelty algorithm was applied to the the seascape data. Figures 7-5 and 7-6 show the distribution of clutter, and poorly-segmented object, novelty ( $\hat{p}(\mathbf{x})$ ) values for this database, and for different types of feature <sup>3</sup>. Both sets of Figures demonstrate the classification rates possible with the novelty classifier (only (EX0 IN3) data used in the latter.) This was done by using the standard Bayes rule, coupled with the class conditional,  $k$ -NN probability density estimates, for the three defined seascape classes. If  $p(\mathbf{x})$  fell below a set threshold for a particular test example, then the object was classified as the fourth, novel, class. The Figures show the effect on classification by varying this threshold. It must be noted that these classifications were only valid for the particular clutter prior probability implied by the population of the test set. In reality this probability would include temporal fluctuations, caused by environment or system dynamics, such as a change in segmentation parameters. Consequently, choosing a threshold that minimised misclassification according to these plots was only correct in this particular case. If the probability of clutter increased then the novelty threshold would need to be increased.

Each of the feature sets demonstrated the ability to successfully separate out the clutter. The maximum classification performance was also similar to the  $C + 1$  rate. The results based on the poorly-segmented data were more interesting.

Figure 7-6 shows that objects with only slight segmentation failure were difficult to identify. The features were not preserving the artifacts that identified these faults. This was not surprising as the features were not designed for this function. The surprising result was the poor

---

<sup>3</sup>The novelty distributions were not weighted by their prior probabilities



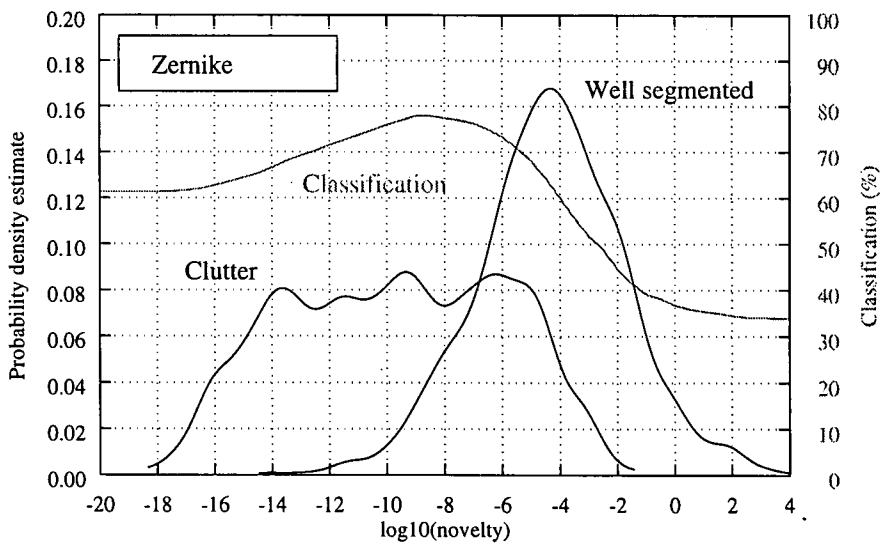
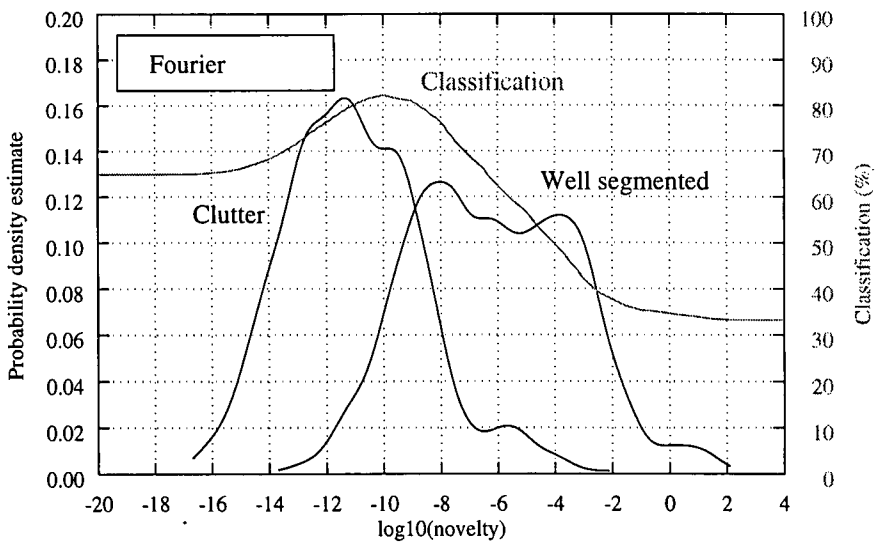
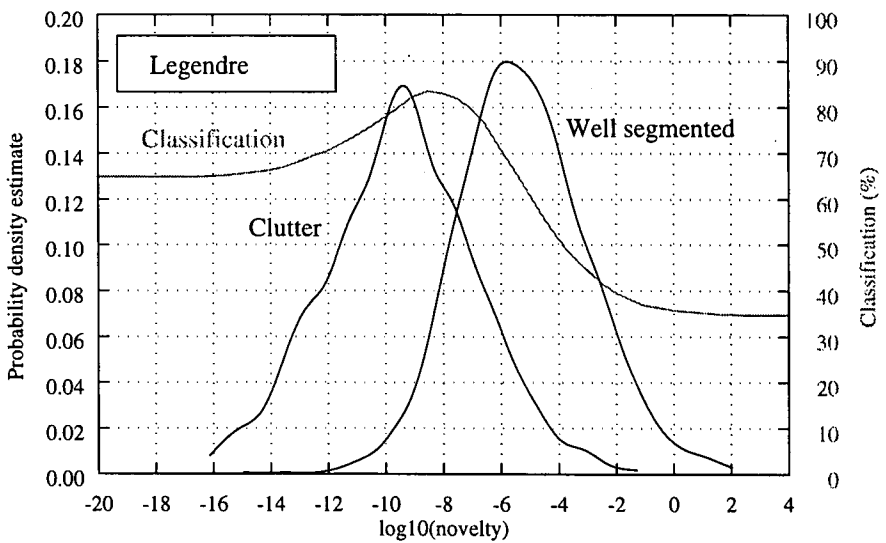
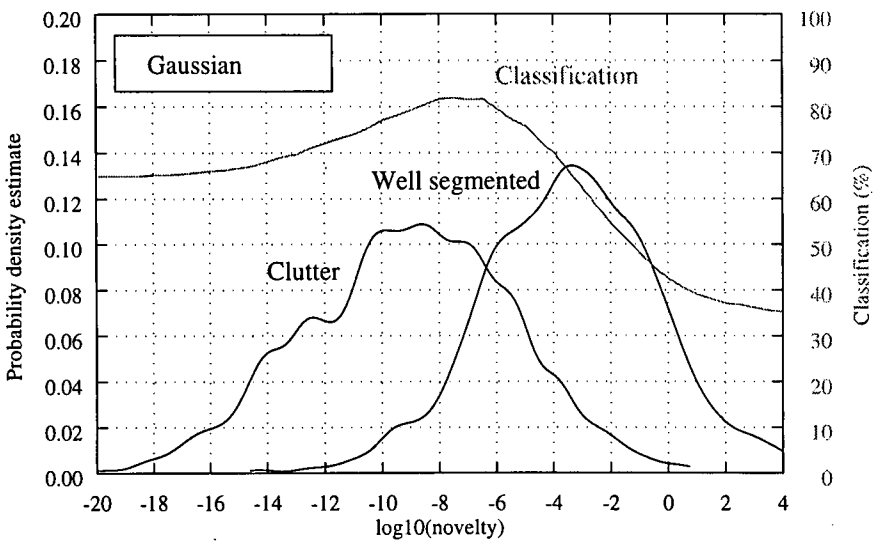


Figure 7-5: Seascap: Distribution of novelty values for both non-rogue and clutter data.

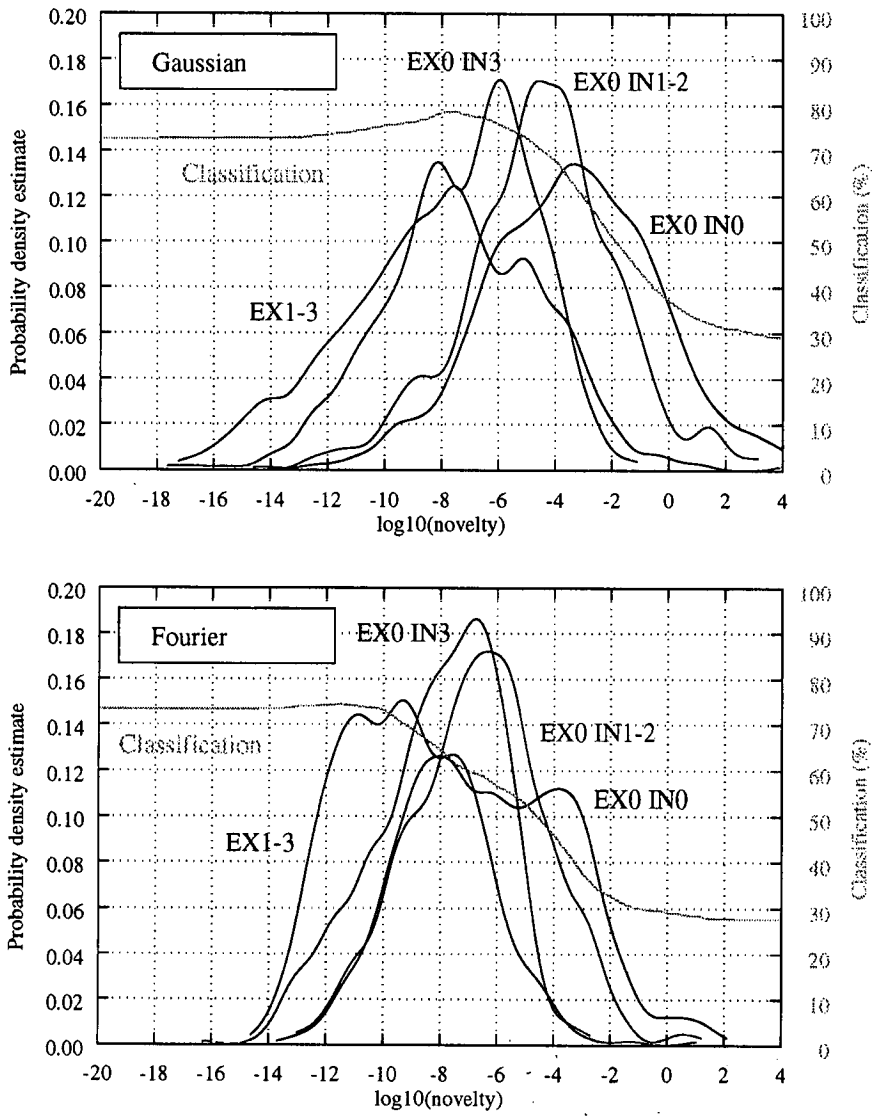
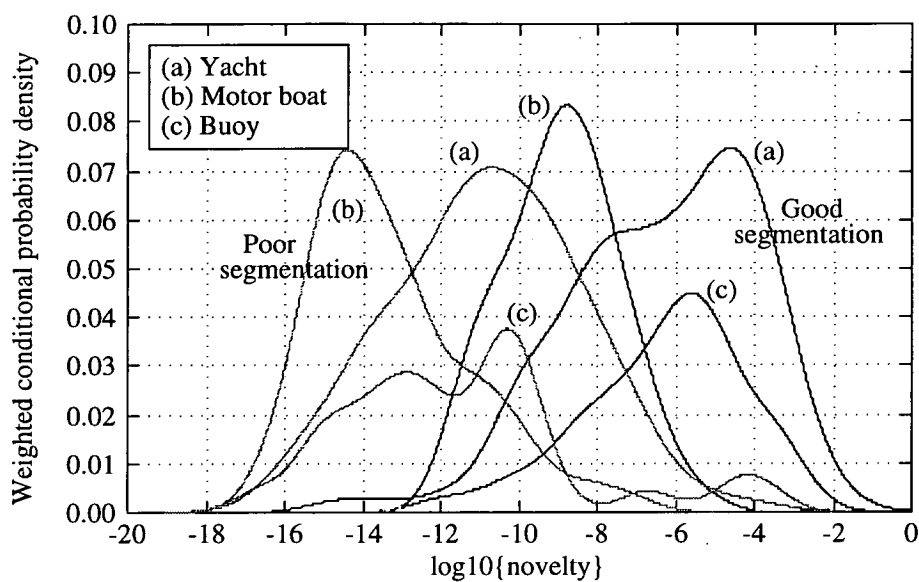


Figure 7-6: Seascope: Distribution of novelty values for both non-rogue and poorly-segmented data.



**Figure 7-7:** Seascape: Novelty distribution of Fourier features of 3 classes.

performance of the Fourier features, and the excellent results using the Zernike data. This was not at all expected from the  $C + 1$  classifications. The rogue Fourier data was separable, so why were the points not suitably classified? The problem was that the data available for test was not uniformly distributed, and was, as has already seen, localised. The distribution was close to the large non-rogue sailboat data which gave them relatively large novelty values. The motor boat Fourier feature distribution, on the other hand, were both weighted by a small prior probability and of high variance. These contributed to produce small novelty values, even smaller than the rogue data. This is demonstrated in Figure 7-7 where the main mass of the small non-rogue data is coming from the motor boats. Further verification was provided by the confusion matrices for varying thresholds in Tables 7-8 and 7-9. Only the (EX0 IN3) rogue data points were used. The failures were transferred, as the threshold increases, from rogue data classified as sailboats directly to motor boats as rogue data.

Guess	Correct class			
	Sail	Motor	Buoy	Novel
Sail	710	4	6	454
Motor	1	513	3	32
Buoy	23	7	329	49
Novel	4	9	0	33
Total	738	533	338	568

(a)  $\log_{10}(\text{novelty}) = -12$  (72.75% correct)

Guess	Correct class			
	Sail	Motor	Buoy	Novel
Sail	704	4	6	438
Motor	1	479	3	18
Buoy	23	7	329	44
Novel	10	43	0	68
Total	738	533	338	568

(b)  $\log_{10}(\text{novelty}) = -11$  (72.5% correct)

**Table 7–8.** Seascape: Confusion matrices for novelty classifier using Fourier features.

Guess	Correct class			
	Sail	Motor	Buoy	Novel
Sail	697	4	5	406
Motor	1	420	2	7
Buoy	23	7	328	37
Novel	17	102	3	118
Total	738	533	338	568

(a)  $\log_{10}(\text{novelty}) = -10$  (71.75% correct)

Guess	Correct class			
	Sail	Motor	Buoy	Novel
Sail	683	3	5	364
Motor	0	289	1	2
Buoy	23	6	324	29
Novel	32	235	8	173
Total	738	533	338	568

(b)  $\log_{10}(\text{novelty}) = -9$  (67.5% correct)

**Table 7–9.** Seascape: Confusion matrices for novelty classifier using Fourier features.

The Zernike features, conversely, mapped the (EX0 IN3) data such that their novelty values were significantly less than any of the non-rogue data values. This could not be predicted, and this problem was caused by multi-modal  $p(\mathbf{x})$  distributions and could not be addressed within the time scope of this thesis.

7.2.4 Identification conclusions

The novelty classifier, though required when the rogue data distributions were unknown, had several disadvantages.

- The novelty classifier required an estimate of a multidimensional probability density function.
- Most approaches, including  $k$ -NN, are fundamentally non-parametric in that they require storage of many training samples.
- The classifier was not suited for integration of the adaptive networks discussed in this thesis, unlike the  $C + 1$  classifier.
- Required setting a threshold through experimentation. As the system will adapt this will need to adapt with it, although as previously stated using a large threshold only means extra processing by a secondary, possible re-segmentation, module. The extra processing may effect response time which will be disadvantageous in an hostile environment.
- It was difficult to choose a suitable set of features that not only differentiated the main classes, but also successfully identified rogue data. This is nonsensical if the rogue data is truly novel.

### 7.3 The separation of rogue data

The previous section demonstrated that it was possible separate out, quite effectively, the rogue data. It was also possible to identify between the two types of rogue data: clutter, and poorly-segmented objects; although the misclassification rate was approximately 30%. This identification was useful as clutter could simply be ditched, and the segmentation failures sent on for further processing, such as re-segmentation. However, the choice of feature was critical, and unfortunately as has been stated can often not be predicted in advance, as rogue data by definition may only appear at run-time.

## 7.4 Database adaptability

In the thesis it was stated that one of the objectives of the work was to design a *plug-and-play* classification module. For each new application into which the classifier is to be used the module should be ready to run after a simple set-up procedure. This entails exposing the system to a labelled, well-segmented, database indicative of the objects that the system will encounter.

With the exception of the NIST database, only one real-world database was used to test the adaptive feature extraction and classification models. A second database was available from BASE<sup>4</sup> for the testing of the system. This database had never been tested before. If the adaptive networks could learn to classify these very different objects as successfully as the seascape data it would further support the flexible, and generic, nature of these type of classifiers, which is one of their strongest attributes.

This new car database was chosen, in particular, for various reasons. It was another real infrared database<sup>5</sup> but with different characteristics. The objects were much similar than in the seascape database and, thus, potentially harder to discriminate. However, the segmentation was easier as the cars were very hot in relation to their surroundings. The surroundings generated their own problems, yet again different to the seascape data and this included swaying trees, the number of car occupants, the direction of the car, as well as differing weather conditions and times of day. The seascape data was all taken under constant weather conditions. The next sections further describes the nature of the car database.

### 7.4.1 Database description

The database contained four classes of vehicle: a Range rover, a Rover car, Ford Fiesta and a Maestro. These were captured, as with the seascape, using a thermal infrared sensor with one vehicle per frame at a constant viewing angle. The resulting images were 512x512, 8 bit per pixel, frames of data. The vehicles themselves were captured at a distance of approximately 50

---

<sup>4</sup>Courtesy of Andy Connelly, University of Edinburgh.

<sup>5</sup>A different sensor would have been more appropriate but was not available at the time.

metres and typically were 64x64 pixels in size. The vehicles were then segmented out of the frames and normalised to 32x32 pixel images, exactly as with the seascape data. Table 7–10 provides the distribution of the classes.

Class	Population
Range Rover	496
Rover car	689
Ford Fiesta	695
Maestro	883
Total	2763

**Table 7–10:** Car: Class distributions.

7.4.2 Database results

The object database was split into training, testing and validation sets. Fixed features were derived and classified for reference. Both linear and nonlinear adaptive feature extraction models were then applied to the data. The mean classification percentages, over 10 tests, for the fixed features are given in Tables 7–11. Standard deviations are given in brackets.

Feature	Linear	7-NN
Legendre	78.0 (1.7)	92.4 (1.2)
Fourier	79.2 (1.5)	95.4 (0.6)

**Table 7–11:** Car: Results for the infrared vehicle data using fixed features. Each score is the mean percentage classification over 10 different samples each consisting of 500 test vectors. The value in parentheses is the standard deviation over the 10 tests.

Results for the adaptive models are given in Table 7–12. Four parameter  $(x_0, y_0, a, b)$  Gaussian kernels were used due to their success with the seascape data. Table 7–13 provides the confusion matrix from one linear, and one nonlinear, adaptive classification test.

Type	Kernels	Hidden units	Classification
Linear	9	-	73.6 (1.0)
Linear	12	-	79.6 (1.3)
Nonlinear	9	12	96.2 (1.4)
Nonlinear	12	8	95.8 (1.1)

**Table 7–12:** Car: Results for the infrared vehicle data using adaptive features. Each score is the mean percentage classification over 10 different samples each consisting of 500 test vectors. The value in parentheses is the standard deviation over the 10 tests.

Guess	Correct class				Guess	Correct class			
	Land	Rover	Fiesta	Maestro		Land	Rover	Fiesta	Maestro
Land	80	6	3	1	Land	98	9	0	0
Rover	8	63	0	10	Rover	3	106	3	1
Fiesta	8	19	107	5	Fiesta	0	4	117	1
Maestro	5	31	10	144	Maestro	0	0	0	158

(a) Linear (78.8% correct)

(b) Nonlinear (95.8% correct)

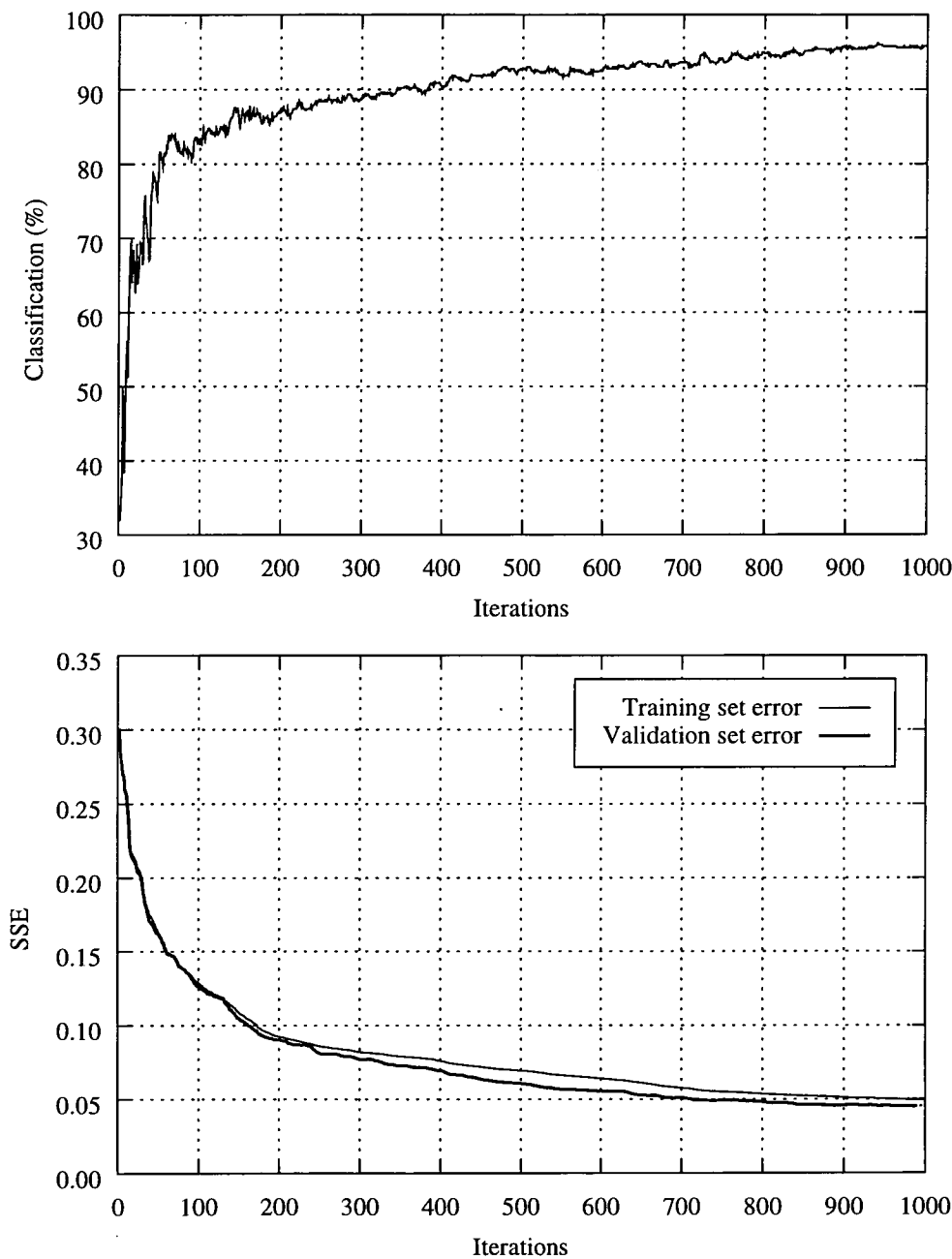
**Table 7–13:** Car: Confusion matrices for the linear and nonlinear adaptive classifiers.

Figure 7–8 shows how the validation set classification rate, the validation set error and the training set error changed during optimisation. This demonstrates that the optimisation path through the error surface was smooth.

7.4.3 Database conclusions

The adaptive models performed well on this completely new database, scoring classification rates in excess of the fixed feature results. Admittedly only two fixed features were tested but this does not distract from the point that excellent generalisation was achieved with adaptive models without any extensive investigation into other features. An object database was created, analysed for properties such as segmentation quality, and applied to four configurations of the adaptive models. No other work was required.





**Figure 7–8:** Car: Validation set classification rate, validation set error and training set error during optimisation for the nonlinear 9 kernel, 12 hidden unit adaptive model.

## 7.5 Review

This chapter has examined the effects on classification performance due to the non-ideal nature of the object generating process. The problems with the BASE data was suggested to be symptomatic of the more generic nature of object generation. The BASE data being simply one example of the distribution of rogue data. As such, two methods were proposed in order to identify these type of objects based on assumptions on the nature of the rogue data distribution. The  $C + 1$  algorithm was found to be far more appropriate if the object generating process were to remain constant and if this fixed rogue data distribution could be adequately sampled. If this assumption is correct then the rogue data can be treated identically to the object data. The success of the classification being dependent only on the overlap of class conditional feature distributions. The  $C + 1$  algorithm had another very important benefit in that it was able to use the adaptive models introduced in the thesis to provide, again, improved classification over many of the standard approaches.

The second approach using novelty detection was found to be superior in situations where the object generating process was known to fluctuate or where the rogue data distributions were heavily undersampled during model optimisation. In these cases the  $C + 1$  algorithm will be dependent on where the rogue data distributions shift or appear. The novelty approach could not, however, directly incorporate the adaptive feature extraction methods.

In order to demonstrate the adaptability of the algorithms discussed in the thesis, the adaptive classification models were applied to a completely new real-world database. With this new database classification rates comparable and better than the standard approaches were achieved.

---

# Chapter 8

## Conclusions

---

This final chapter summarises the work which has been carried out, reviews the extent to which the aims, set down in Chapter 1, have been achieved, and indicates where future work could most productively be focussed.

### 8.1 Summary of work completed

Chapters 3 through 7 were a chronological, and systematic, record of the work completed during this project. These chapters represent the solutions to the aims that were outlined in Chapter 1. The work was modularised such that each chapter centred on a particular set of aims. Chapter 3 analysed the generation of two real, non-ideal, databases derived from a set of infrared images. Chapter 4 looked at applying standard feature extraction and classification techniques to the accurately-generated data of Chapter 3, and discussed the complexities of such methods. Chapter 5 introduced the successful application of a relatively new classification model to the real data, simplified the model and then extended it to improve performance even more. Comparisons were made in this chapter with the results of the standard approaches. Incorporation of invariance in the new model was described in Chapter 6, and this invariance was also tested with the real data. Chapter 7 examined the effects of a non-ideal preprocessing system, something that Chapter 3 highlighted, and demonstrated both the effect and remedy with respect to the new classification model. This tested the improved classification module with even more realistic data. A final test in the chapter was to apply the new techniques to a completely new database to show that the system was easily adaptable to a new environment. The next section explains how far the original aims were achieved.

## 8.2 Analysis of completed aims

### 1. *To highlight problems with the existing BASE ATR object classification system.*

In Chapter 2 the problems with the existing BASE ATR classification module were discussed. There was little consideration of the data being classified including object characteristics, preprocessing or generation. No thought was given to the required solution, to the scale or type of the classification model used, and inappropriate model parameter estimation procedures were applied. These issues were addressed at various points in this thesis.

### 2. *To design a replacement classification module for the BASE ATR system.*

The combined feature extraction and classification model described in this thesis, in its nonlinear form, is structurally equivalent to the original BASE MLP classifier; an image input layer, followed by a nonlinear hidden layer, plus a linear output layer. In fact, the only difference is the number of hidden units, and the values of the model parameters. No further storage was thus required, and throughput has not changed, with the exception of some preprocessing that is not currently performed in the BASE system. Furthermore, the number of *adaptive* parameters has considerably decreased from the order of thousands to about one hundred.

The ATR module generated, consisting of the combined feature extraction and classification model, has been tested in the OSTRICH ATR system, and is readily available to BASE. The module has been tested on two real-world ATR databases captured using an infrared sensor. No testing has been performed with any other type of sensor due to time restrictions.

### 3. *To provide improved classification.*

Chapters 5 and 6 have shown that the new combined models can provide increased, or at least, equivalent classification performance than the standard approaches. In fact, some of the linear new model results outperformed the nonlinear classification of certain standard

features. This implies good classification results without the need for any nonlinear calculations.

In the cases where the classification results were equivalent, for example with the nonlinear discrimination of the selected Fourier features on the seascape data, the differentiation between the two approaches was the amount of time taken to achieve the same result.

#### 4. *To design a classifier that is adaptive to new environments and applications.*

The ability of the new adaptive models to learn new environments and applications is one of the most important successes of the thesis. In this thesis the adaptive models were applied to three different databases; two real-world infrared applications with different environmental factors and clutter sources, and a completely unsimilar database of handwritten characters. In each case the classification results were equivalent, if not better, than any of the laborious approaches using the traditional two-stage and separate feature extract and classify approach.

In each of the problems once the data had been generated it was a matter of optimising the model with the new data using a suitable number of kernels, and hidden units. There was no need for complicated feature extraction, or selection, procedures, as the features best suited for classification are automatically generated. Of course, the features with these new models will be confined to those generated by the linear, spatial mappings of the image object data, where the transformation can only exist in the set of all possible manifestations of the finite sum of kernels.

As stated in Chapter 1 these new adaptive models inherit the disadvantages of all segment-feature-classify approaches. The most important is that the classification can only be made with reference to the image data presented, with all important range, temporal and contextual information removed at this point. Also, as stated in the last paragraph the features generated are restricted to those generated by the linear kernel mapping. Both combined means that the susceptibility of these models to decoys is high. However, there are two points that need to be expressed at this point. Firstly, decoys could potentially be included in the estimation of adaptive models meaning the decoys would have to be detailed to fool the classifier. Secondly, and more importantly, and as stated in Chapters 1 and 2, these systems tend to be used in conjunction with tracking systems, knowledge

bases, and model matching systems, all with access to multiple sensors. The purpose of the adaptive models is to classify an object based solely, and as best it can without bias from other sources of information, on the object data shape. It is the job of later interpretation stages to examine the evidence from *all* parts of the system to make a final reasoned judgement based on the generated results.

A final note on the adaptive nature of the combined model is that the structure of the model is identical to many neural classifiers already implemented in many ATR systems, including BASE. The only difference being the estimation of the model and, potentially, the number of processing nodes. This implies that the adaptive models could be incorporated immediately into already installed classification systems by simply updating the model weights and biases.

Estimation of the model parameters was performed directly against the object image data, and so no complicated feature extraction or selection techniques were required. A conjugate gradient line-searching technique allowed for faster optimisation of the model, with fewer control values required to be set. The only other control values needed to be set were the number and type of feature extracting kernel, although the many-simple-kernel approach seemed to work well in all cases, and for the nonlinear version of the model, the number of hidden units. The model validation procedure, however, remains the most difficult operation, and care must be taken in determining when to stop optimisation. However, this is a general model estimation problem. A last point is that these adaptive models, with the higher dimensional inputs, have more computations to perform, compared with, for example, a 20 input standard MLP and thus it can take longer to generate an optimised model.

##### *5. To analyse the real data provided for the project.*

There are several reasons for performing analysis on real, or synthetic, data before any classification procedures are applied. These were described at the beginning of Chapter 3. In Chapter 3, the real, seascape, data that was provided for the project was analysed. In particular, the method of generating and preprocessing the data, and any assumptions used in these processes. The object characteristics were also considered. This provided very useful information for determining features for discriminating between the classes of object, as well as, for reasoning classification successes and failures.

6. *To incorporate invariance to size, position, or two-dimensional rotations of the object image, into the classification model.*

The incorporation of invariance was discussed in Chapter 6. Two alternative methods, both that used a single feature extraction and classification model, were proposed. One model was found to have problems in parameter estimation. A solution was found to this problem but it was felt that the approach was still inappropriate. A better proposal was to use a preprocessing step that introduced the required invariance into the data, as opposed to the model itself, and allowed the application of the previously successful techniques when no invariance had been available. Adding the invariance led to reduced classification performance with the real data. This was expected as orientation was a very important feature in classifying the real, seascape, database.

7. *To identify potential weakness in the new classification module and the identification of rogue data.*

Throughout the testing of both the standard and the new model classification, tests on both the NIST and seascape databases confusion matrices were provided that showed where the main sources of misclassification were occurring. For the seascape databases, as was noted in the preliminary analysis of the data in Chapter 3, the main confusions existed between the sailboats and buoys, and between the motor boats and clutter. The clutter was one type of rogue data generated by the non-ideal segmentation module. This inability of the new classification models to handle this rogue data was a potential weakness. Chapter 7 examined how this weakness could be overcome.

In Chapter 7 two approaches to the detection of rogue data was discussed. If the distribution of the rogue data was constant and well sampled then a simple  $C + 1$  classification method could be used in conjunction with the adaptive feature extraction and classification models to generate results potentially dependent only on the amount of distribution overlap between all the object and rogue data classes. This is a powerful method for rogue data detection, and could even potentially incorporate Bayes risk methods for reducing the effects of rejecting object data as rogue data, or vice-versa.

One potential weakness of the new classifiers is that if the assumption of the rogue data does not hold, if, for example, the segmentation process is adaptive. The  $C + 1$

algorithm could potentially fail disastrously with rogue data, as although the data may still be dissimilar to object data, it will be classified as objects due to the location in feature space. A novelty approach was discussed for countering this shift in the rogue data distributions but unlike the  $C + 1$  approach it could not incorporate the benefits of the adaptive feature extraction classifiers.

A further potential weakness of the new model, as already discussed, is that it is not 100% guaranteed to find the best feature set for classification. These may be caused by local minima in the optimisation process of the model, not reaching the global error minima, but is more likely to be caused by the nature of the feature extraction mechanism. The linear mapping of the object data used to generate the features may not be able to approximate the best mapping of the data due to the form of the kernel. Furthermore, the best feature extraction may not be even expressible as a linear mapping of the image data.

### 8.3 Scope for future work

There are many avenues of research that were not attempted or published here, due to either lack of relevance to the thesis or time to complete. Some are listed on the following page, in no particular order.



- The effect of preprocessing on the classification results. Could this step be simplified, without adversely affecting classification performance?
- Identification of subclasses in the seascape database. This was not attempted due to the lack of examples in many of the classes.
- Examining how classification of an object changes with the new model as it is tracked and rotates out of the image plane.
- What is the minimum size of object that can be extracted, and confidently classified.
- Producing *a posteriori* classification results. This was performed but there was no time to report results.
- For the combined model, only a few types of kernels were tried, and all based on a Gaussian mother kernel. In multivariate kernel density estimation it is known that Gaussian kernels are not the most efficient [103]. It would be interesting to examine other types of kernels.
- More research is required on the apparent over- and under-fitting that occurred with the new model in Chapter 5, as well as, initialising the models before optimisation.
- Examining the effect of using a risk-based classification criterion.
- The application of Bayesian inference techniques which have recently become popular for determining neural network models has also been neglected in this thesis. This could incorporate investigation into using better validation techniques for estimating generalisation, such as bootstrapping.

## 8.4 Final comment

Although not guaranteed to find the best features for classifying, and the fact they take longer to optimise than a standard MLP, the combined feature extraction and classification model, together with the invariance introduced by preprocessing the data, offers a very suitable model for an ATR classification module. The model offers ease-of-use, easy adaptability to new environments, and typically good generalisation.

---

## References

---

- [1] Y. S. Abo-Mostafa and D. Psaltis. "Recognitive Aspects of Moment Invariants". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):698–706, November 1984.
- [2] P. R. Aday and M. A. H. Dempster. "Introduction to Optimization Methods". Chapman and Hall, 1978.
- [3] H. C. Andrews. "Multidimensional Rotations in Feature Selection". *IEEE Transactions on Computers*, C-20(9):1045–1051, September 1971.
- [4] N. R. Pal and S. K. Pal. "A review on image segmentation techniques". *Pattern Recognition*, 26(9):1277–1294, 1993.
- [5] H.B. Barlow. "Summation and inhibition in the frogs retina". *J. Physiol. (Lond.)*, 119:69–88, 1953.
- [6] E. Barnard and D. Casasent. "Invariance and neural nets". *IEEE Transactions on Neural Networks*, 2(5):498–508, 1991.
- [7] E. M. L. Beale. "Introduction to Optimization". Wiley and Sons, 1988.
- [8] R. Beale and T. Jackson. "Neural Computing". Adam Hilger, 1990.
- [9] S. O. Belkasim, M. Shridhar, and M. Ahmadi. "Pattern Recognition with Moment Invariants: A Comparative Study and New Results". *Pattern Recognition*, 24(12):1117–1138, 1991.
- [10] B. Bhanu. "Automatic Target Recognition: State of the Art Survey". *IEEE Transactions on Aerospace and Electronic Systems*, AES-22(4):364–379, July 1986.
- [11] A. B. Bhatia and E. Wolf. "On the circle polynomials of Zernike and related orthogonal sets". *Proceedings of the Cambridge Philosophical Society*, 50:40–48, 1954.
- [12] C. Bishop. "Neural Networks for Pattern Recognition", chapter 7, pages 253–294. Clarendon Press, 1995.
- [13] C. M. Bishop. "Neural Networks for Pattern Recognition". Clarendon Press, 1995.
- [14] C.M. Bishop. "Novelty detection and neural network validation". *IEE Proceedings: Vision, Image and Signal Processing*, 141(4):217–222, 1994.
- [15] G. Bradski and S. Grossberg. "Fast-Learning VIEWNET Architectures for Recognizing Three-dimensional Objects from Multiple Two-dimensional Views". *Neural Networks*, 8(7/8):1053–1080, 1995.

- [16] D. S. Broomhead and D. Lowe. "Multivariate Functional Interpolation and Adaptive Networks". *Complex Systems*, 2:321–355, 1988.
- [17] W. M. Brown and C. W. Swonger. "A Prospectus for Automatic Target Recognition". *IEEE Transactions on Aerospace and Electronic Systems*, 25(3):401–409, May 1989.
- [18] D. Casasent. "General-Purpose Optical Pattern Recognition Image Processors". *Proceedings of the IEEE*, 83(11), November 1994.
- [19] D. Casasent, J. Smokelin, and A. Ye. "Wavelet and Gabor transforms for detection". *Optical Engineering*, 31(9):1893–1896, 1992.
- [20] D. P. Casasent and J. Smokelin. "Neural net design of macro Gabor wavelet filters for distortion-invariant object detection in clutter". *Optical Engineering*, 33(7):2264–2271, 1994.
- [21] D. P. Casasent and J. Smokelin. "Real, imaginary, and clutter Gabor filter fusion for detection with reduced false alarms". *Optical Engineering*, 33(7):2255–2263, 1994.
- [22] B. Cheng and D.M. Titterington. "Neural Networks: A Review from a Statistical Perspective". *Statistical Science*, 9(1):2–54, 1994.
- [23] A. Connelly. "Temporal Aspects of Classification: Database Preparation". Tn4317, British Aerospace Systems and Equipment, 1995.
- [24] T. M. Cover and P. E. Hart. "Nearest Neighbour Pattern Classification". *IEEE Transactions on Information Theory*, IT-13(1):21–27, January 1967.
- [25] C. Daniell, D. H. Kemsley, W. P. Lincoln, W. A. Tackett, and G. A. Baraghimian. "Artificial neural networks for automatic target recognition". *Optical Engineering*, 31(12):2522–2531, December 1992.
- [26] J. G. Daugman. "Complete Discrete 2-D Gabor Transforms by Neural Networks for Image Analysis and Compression". *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(7):1169–1179, 1988.
- [27] J. G. Daugman. "Uncertainty relation for resolution in space, spatial frequency, and orientation optimised by two-dimensional visual cortical filters". *Journal of the Optical Society of America (A)*, 2(7):1160–1169, July 1985.
- [28] P. A. Devijver and J. Kittler. "*Pattern Recognition: A Statistical Approach*". Prentice-Hall International, 1982.
- [29] R. J. Drazovich, F. X. Lanzinger, and T. O. Binford. "Radar target classification". In *Proceedings of the IEEE Conference on PRIP*, pages 496–501, 1981.
- [30] R. Duda and P. Hart. "*Pattern Recognition and Scene Analysis*". Wiley and Sons, New York, 1941.
- [31] S. A. Dudani, K. J. Breeding, and R. B. McGhee. "Aircraft Identification by Moment Invariants". *IEEE Transactions on Computers*, C-26(1):39–45, January 1977.

- [32] P. J. Edwards. “*Analogue Imprecision in MLPs - Implications and Learning Improvements*”. PhD thesis, Department of Electrical Engineering, University of Edinburgh, 1994.
- [33] M. Ferraro and T. M. Caelli. “Relationship between integral transform invariances and Lie group theory”. *Journal of the Optical Society of America A*, pages 738–742, 1988.
- [34] R. A. Fisher. “The use of multiple measurements in taxonomic problems”. *Annals of Eugenics*, 7:179–188, 1936.
- [35] R. Fletcher and C. M. Reeves. “Function minimization by conjugate gradients”. *The Computer Journal*, 7:149–154, 1965.
- [36] W. T. Freeman and E. H. Adelson. “The Design and Use of Steerable Filters”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.
- [37] J. H. Friedman. “Multivariate Adaptive Regression Splines”. *Annals of Statistics*, 19(1):1–67, 1991.
- [38] M. Fukumi, S. Omatu, and Y. Nishikawa. “Rotation Invariant Neural Pattern Recognition System which Can Estimate A Rotation Angle”. *IEEE Proceedings of the International Conference on Neural Networks*, 7:4390–4395, 1994.
- [39] M. Fukumi, S. Omatu, F. Takeda, and T. Kosaka. “Rotation-Invariant Neural Pattern Recognition System with Application to Coin Recognition”. *IEEE Transactions on Neural Networks*, pages 272–279, 1992.
- [40] K. Fukunaga. “*Introduction To Statistical Pattern Recognition*”. Academic Press, 1972.
- [41] K. Fukushima and S. Miyake. “Neocognitron: A new algorithm for pattern recognition tolerant to deformations and shifts in position”. *Pattern Recognition*, 15(6):455–469, 1982.
- [42] D. Gabor. “Theory of communication”. *Journal of the Institute of Electrical Engineers*, 93:429–457, 1946.
- [43] C. L. Giles, R. D. Griffin, and T. Maxwell. “Encoding Geometric Invariances in Higher-Order Neural Networks”. In D.Z. Anderson, editor, *Neural Information Processing Systems*, pages 301–309, New York, N.Y., 1988. American Institute of Physics.
- [44] C. L. Giles and T. Maxwell. “Learning, Invariance, and Generalization in High-Order Neural Networks”. *Applied Optics*, 26(23):4972, 1987.
- [45] M. H. Glauberman. “Character recognition for business machine”. *Electronics*, pages 132–136, 1956.
- [46] R. C. Gonzalez and R. E. Woods. “*Digital Image Processing*”. Addison-Wesley Publishing, 1992.
- [47] J. W. Goodman. “*Introduction to Fourier Optics*”. McGraw-Hill Inc, 1968.
- [48] M. A. Green. “Neural Networks Classifier Demonstrator: System Study Report”. Tn3650, British Aerospace Systems and Equipment, 1992.

- [49] M. A. Green. "Target Classification with a Neural Network: Final Report". Tn3627, British Aerospace Systems and Equipment, 1992.
- [50] M. A. Green. "Target Classification with a Neural Network: Proposal for 1992". Tn3614, British Aerospace Systems and Equipment, 1992.
- [51] S. Grossberg, H. Hawkins, and A. Waxman. "Introduction: 1995 Special Issue on Automatic Target Recognition". *Neural Networks*, 8(7/8):1003, 1995.
- [52] D. J. Hand. "*Discrimination and Classification*". Wiley and Sons, 1981.
- [53] R. M. Haralick, K. Shanmugam, and I. Dinstein. "Textural Features for Image Classification". *IEEE Transactions on Systems, Man and Cybernetics*, SMC-3(6):610–621, November 1973.
- [54] J. Hertz, A. Krogh, and R. G. Palmer. "*Introduction to the theory of neural computation*". Addison-Wesley Publishing, 1991.
- [55] M. I. Heywood and P. D. Noakes. "Fractional central moment method for movement-invariant object classification". *IEE Proceedings - Vision Image and Signal Processing*, 142(4):213–219, August 1995.
- [56] Y. Ho and A.K. Agrawala. "On Pattern Classification Algorithms - Introduction and Survey". *IEEE Transactions on Automatic Control*, AC-13(6):676–689, December 1968.
- [57] M. Hu. "Pattern Recognition by Moment Invariants". *Proceedings of the IRE*, 49:1428, September 1961.
- [58] A. K. Jain. "*Fundamentals of Digital Image Processing*". Prentice-Hall International, 1989.
- [59] A. K. Jain. "*Fundamentals of Image Processing*", chapter 9, pages 407–414. Prentice-Hall Inc, 1989.
- [60] S. Kadambe and P. Srinivasan. "Applications of adaptive wavelets for speech". *Optical Engineering*, 33(7):2204–2211, 1994.
- [61] A. Khotanzad and Y. H. Hong. "Invariant Image Recognition by Zernike Moments". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(5):489–487, May 1990.
- [62] A. Khotanzad and J. Lu. "Classification of Invariant Image Representations using a Neural Network". *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(6):1028–1038, June 1990.
- [63] C. M. Kocur, S. K. Rogers, L. R. Myers, T. Burns, M. Kabrisky, J. W. Hoffmeister, K. W. Bauer, and J. M. Steppe. "Using Neural Networks to Select Wavelet Features for Breast Cancer Diagnosis". *IEEE Engineering in Medicine and Biology*, 15(3):95–102, 1996.
- [64] A. Krogh and J. Hertz. "A simple weight decay can improve generalisation". In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 950–957, San Mateo, CA, 1992. Morgan Kaufmann publishers.

- [65] L. E. Lazofson. "A biologically inspired neural network architecture for image processing". Master's thesis, Air Force Institute of Technology, 1990.
- [66] M. Levine. "Feature Extraction: A Survey". *Proceedings of the IEEE*, 57(8):1391–1407, 1969.
- [67] Y. Li. "Fourier-Mellin Transform and the Invariant Image Moments". *Japanese Journal of Applied Physics*, 30(7):1045–1046, July 1991.
- [68] R. P. Lippmann. "Pattern Classification Using Neural Networks". *IEEE Communications Magazine*, pages 47–64, November 1989.
- [69] D. J. C. MacKay. "*Bayesian methods for backpropagation networks*", chapter 6. Springer-Verlag, 1994.
- [70] Y. Mallet, D. Coomans abd J. Kautsky, and O. de Vel. "Classification Using Adaptive Wavelets for Feature Extraction". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1058–1066, 1997.
- [71] K. V. Mardia. "*Statistics of Directional Data*". Academic Press, 1972.
- [72] T. Masters. "*Signal and Image Processing with Neural Networks*". Wiley and Sons, New York, 1994.
- [73] G. Nagy. "State of the Art in Pattern Recognition". *Proceedings of the IEEE*, 56(5):836–864, 1968.
- [74] R. M. Neal. "*Bayesian Learning for Neural Networks*". Springer, 1996.
- [75] J. A. Nelder and R. Mead. "A simplex method for function minimization". *The Computer Journal*, 7:308–313, 1965.
- [76] P. R. Norton. "Infrared image sensors". *Optical Engineering*, 30(11):1649–1663, 1991.
- [77] J. A. Parker, R. V. Kenyon, and D. E. Troxel. "Comparison of Interpolating Methods for Image Resampling". *IEEE Transactions on Medical Imaging*, MI-2(1):31–39, 1983.
- [78] S. J. Perantonis and P. J. G. Lisboa. "Translation, Rotation, and Scale Invariant Pattern Recognition by Higher-Order Neural Networks and Moment Classifiers". *IEEE Transactions on Neural Networks*, 3(2):241–251, March 1992.
- [79] L. I. Perlovsky, J. A. Chernick, and W. H. Schoendorf. "Multi-sensor ATR and Identification of Friend or Foe Using MLANS". *Neural Networks*, 8(7/8):1185–1200, 1995.
- [80] E. Persoon and K. Fu. "Shape Discrimination Using Fourier Descriptors". *IEEE Transactions on Systems, Man and Cybernetics*, SMC-7(3):170–179, March 1977.
- [81] W. Press, W. Vetterling, S. Teukolsky, and B. Flannery. Conjugate gradient methods in multidimensions. In "*Numerical Recipes in C*", pages 420–425. Cambridge University Press, 1988.

- [82] W. Press, W. Vetterling, S. Teukolsky, and B. Flannery. Downhill simplex method in multidimensions. In *"Numerical Recipes in C"*, pages 408–412. Cambridge University Press, 1988.
- [83] W. Press, W. Vetterling, S. Teukolsky, and B. Flannery. Transformation method: Exponential and normal deviates. In *"Numerical Recipes in C"*, pages 287–290. Cambridge University Press, 1988.
- [84] K. L. Priddy. *"Feature extraction and classification of FLIR imagery using relative locations of non-homogeneous regions with feedforward neural networks"*. PhD thesis, Air Force Institute of Technology, 1992.
- [85] R. Reavy. "Neural Network Assisted Image Segmentation: Segmentation Techniques and Database Preparation". Tn4140, British Aerospace Systems and Equipment, 1994.
- [86] R. Reavy. "Neural Network Assisted Image Segmentation: Resegmentation System Design, Implementation, and Capability Analysis". Tn4280, British Aerospace Systems and Equipment, 1995.
- [87] B. Reischer. "Target tracking methodologies present and future". In *Proceedings of the Workshop on Imaging Trackers and Autonomous Acquisition Applications for Missile Guidance*, pages 156–165, Nov. 19–20, 1979.
- [88] T. H. Reiss. "The Revised Fundamental Theorem of Moment Invariants". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):830–834, August 1991.
- [89] B. Ripley. "Neural Networks and Related Methods for Classification". *Journal of the Royal Statistical Society(B)*, 56(3):409–456, 1994.
- [90] B. D. Ripley. *"Pattern Recognition and Neural Networks"*. Cambridge University Press, 1996.
- [91] M. W. Roth. "Survey of Neural Network Technology for Automatic Target Recognition". *IEEE Transactions on Neural Networks*, 1(1):28–43, March 1990.
- [92] D. E. Rumelhart and J. L. McClelland. *"Parallel Distributed Processing"*. MIT Press, 1989.
- [93] S. Saarinen, R. Bramley, and G. Cybenko. "Ill-conditioning in neural network training problems". *SIAM*, 14(3):693–714, 1993.
- [94] F. A. Sadjadi. "Automatic Target Recognition". *Optical Engineering*, 31(12):2519–2520, December 1992.
- [95] F. A. Sadjadi and M. Bazakos. "A perspective on automatic target recognition evaluation technology". *Optical Engineering*, 30(2):141–146, February 1991.
- [96] F. A. Sadjadi, H. Nasr, H. Amehdi, and M. Bazakos. "Knowledge- and model-based automatic target recognition algorithm adaptation". *Optical Engineering*, 30(2):183–188, February 1991.
- [97] S. R. Sain, K. A. Baggerly, and D. W. Scott. "Cross-Validation of Multivariate Densities". *Journal of the American Statistical Association*, 89(427):807–817, 1994.

- [98] W. S. Sarle. "Neural Networks and Statistical Models". *Proceedings of the Nineteenth Annual SAS Users Group International Conference*, April 1994.
- [99] H. Sawai. "Axially Symmetric Neural Network Architecture for Rotation-Invariant Pattern Recognition". *IEEE Proceedings of the International Conference on Neural Networks (Orlando)*, 7:4253–4258, 1994.
- [100] R.J. Schalkoff. "*Pattern Recognition: Statistical, Stuctural, and Neural Approaches*". Wiley and Sons, 1992.
- [101] H. C. Schau. "Shape recognition with scale and rotation invariance". *Optical Engineering*, pages 268–274, 1992.
- [102] B. W. Scott. "Lantirn system flight testing begins". *Aviation Week and Space Technology*, pages 117–126, 1982.
- [103] D. W. Scott. "*Multivariate Density Estimation: Theory, Practice, and Visualisation*". Wiley and Sons, New York, 1992.
- [104] W. S. Shao and Y. S. Chen. "Pattern Analysis On Shift, Rotation, and Scaling". *Electronic Letters*, pages 2271–2273, 1992.
- [105] Y. Sheng. "Fourier-Mellin spatial filters for invariant pattern recognition". *Optical Engineering*, 28(5):494–500, 1989.
- [106] Y. Sheng and H. H. Arsenault. "Experiments on pattern recognition using Fourier-Mellin descriptors". *Journal of the Optical Society of America A*, 3(6):771–776, June 1986.
- [107] Y. Sheng and J. Duvernoy. "Circular-Fourier-radial-Mellin transform descriptors for pattern recognition". *JOSA communications*, 3(6):885–888, June 1986.
- [108] Y. Sheng and C. Lejeune. "Invariant Pattern Recognition using Fourier-Mellin Transforms and Neural Networks". *Journal of Optics*, 22(5):223–228, 1991.
- [109] Y. Sheng and L. Shen. "Orthogonal Fourier-Mellin moments for invariant pattern recognition". *Journal of the Optical Society of America (A)*, 11(6):1748–1757, 1994.
- [110] R. N. Shepard and J. Metzler. "Mental Rotation of Three-Dimensional Objects". *Science*, 171:701–703, 19 February 1971.
- [111] A. Shustorovich. "A Subspace Projection Approach to Feature Extraction: The Two- Dimensional Gabor Transform for Character Recognition". *Neural Networks*, 7(8):1295–1301, 1994.
- [112] A. Shustorovich and C. W. Thrasher. "Neural Network Positioning and Classification of Handwritten Characters". *Neural Networks*, 9(4):685–693, 1996.
- [113] P. Simard, B. Victorri, Y. Le Cun, and J. Denker. "Tangent Prop - A formalism for specifying selected invariances in an adaptive network". In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 895–903, San Mateo, CA, 1992. Morgan Kaufmann publishers.



- [114] M. Smart. "Noise in neural training and Infra-red Image classification: Database Preparation". TN3938, British Aerospace Systems and Equipment, 1993.
- [115] M. Smart. "Infra-red Image Classification Preliminary Investigation". TN4132, British Aerospace Systems and Equipment, 1994.
- [116] M. H. W. Smart and A. F. Murray. "Multilayer Perceptron for Rotationally Invariant Feature Extraction and Classification". In S. K. Rogers and D. W. Ruck, editors, *Proceedings of the SPIE International Conference on Applications and Science of Artificial Neural Networks II, Florida*, volume 2760, pages 459–466, Bellingham, WA, 1996. SPIE.
- [117] F. W. Smith and M. H. Wright. "Automatic Ship Photo Interpretation by the Method of Moments". *IEEE Transactions on Computers*, C-20:1089–1094, September 1971.
- [118] H. Szu, B. Telfer, and J. Garcia. "Wavelet Transforms and Neural Networks for Compression and Recognition". *Neural Networks*, 9(4):695–708, 1996.
- [119] H. H. Szu, B. Telfer, and S. Kadambe. "Neural network adaptive wavelets for signal representation and classification". *Optical Engineering*, 31(9):1907–1916, 1992.
- [120] H. H. Szu and B. A. Telfer. "Mathematics of adaptive wavelet transforms: relating continuous with discrete transforms". *Optical Engineering*, 33(7):2111–2124, 1994.
- [121] G. L. Tarr. "*Multi-layered feedforward neural networks for image segmentation*". PhD thesis, Air Force Institute of Technology, 1991.
- [122] M. R. Teague. "Image analysis via the general theory of moments". *Journal of the Optical Society of America*, 70(8):920–930, 1980.
- [123] C. Teh and R. T. Chin. "On Digital Approximation of Moment Invariants". *Computer Vision, Graphics and Image Processing*, 33:318–326, 1986.
- [124] C. Teh and R. T. Chin. "On Image Analysis by the Method of Moments". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-10(4):496–513, July 1988.
- [125] B. A. Telfer, H. H. Szu, G. J. Dobeck, J. P. Garcia, H. Ko, A. Dubey, and N. Witherpoon. "Adaptive wavelet classification of acoustic backscatter and imagery". *Optical Engineering*, 33(7):2192–2203, 1994.
- [126] C. Hian-Ann Ting. "Rotation Invariant Neocognitron". *IEEE Proceedings of the International Conference on Neural Networks (Singapore)*, pages 2216–2221, 1991.
- [127] M. Tistarelli and G. Sandini. "On the Advantages of Polar and Log-Polar Mapping for Direct Estimation of Time-to-impact from Optical Flow". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):401–410, 1993.
- [128] Ø. D. Trier, A. K. Jain, and T. Taxt. "Feature Extraction Methods for Character Recognition". *Pattern Recognition*, 29(4):641–662, 1996.

- [129] S. Troxel, S. Rogers, and M. Kabrinsky. "The Use of Neural Networks in PSRI Target Recognition". *Proceedings of the IEEE International Conference on Neural Networks (San Diego)*, pages I-593-I-600, 1988.
- [130] F. Vivarelli and C. K. I. Williams. "Using Bayesian neural networks to classify segmented images". Ncrg/97/007, Neural Computing Research Group, Aston, April 1997.
- [131] P. F. Walker. "Smart weapons in naval warfare". *Scientific American*, 248:53-61, 1983.
- [132] A. M. Waxman, M. C. Seibert, A. Gove, D. A. Fay, A. M. Bernardon, C. Lazott, W. R. Steele, and R. K. Cunningham. "Neural Processing of Targets in Visible, Multispectral IR and SAR Imagery". *Neural Networks*, 8(7/8):1029-1051, 1995.
- [133] D. Weber and D. Casasent. "Fusion and optimized Gabor filter design for object recognition". *Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE)*, 2588(70):662-675, 1995.
- [134] E. J. Wegman. "Nonparametric Probability Density Estimation: I. A Summary of Available Methods". *Technometrics*, 14(3):533-546, 1972.
- [135] H. White. *"Artificial Neural Networks: Approximation and Learning Theory"*. Blackwell, 1992.
- [136] C. L. Wilson and M. D. Garris. "NIST Handprinted Character Database". From <ftp://sequoyah.ncsl.nist.gov/pub/database/data>.
- [137] J. Wood. "Invariant Pattern Recognition: A Review". *Pattern Recognition*, 29(1):1-17, 1996.

---

## Appendix A

# Optimisation techniques

---

There are many approaches to the nonlinear optimisation of adaptive parameters against some error criterion and there exists an extensive literature on the subject [2,7,12]. There exists no optimal optimisation technique and the technique chosen is often dependent on the problem at hand.

Most neural networks optimisation techniques are iterative in nature and make use of zero, first and second order derivatives to determine the position of the local error minimum. Three popular techniques are described in this appendix, of which the last two are the only optimisation techniques used in this thesis. The other optimisation algorithms are generally based on the availability of the error Hessian such as quasi-newton and Levenberg-Marquardt method. The methods used in this thesis are deemed adequate for the task involved.

### A.1 Simple descent methods

Early neural network models, such as the multi-layer perceptron, used a basic optimisation technique known as gradient descent. This involves *back-propagating* errors from the model output to the input and then taking fixed steps in the direction of the local negative error surface gradient i.e.

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \eta \nabla E|_{\mathbf{w}(t)}$$

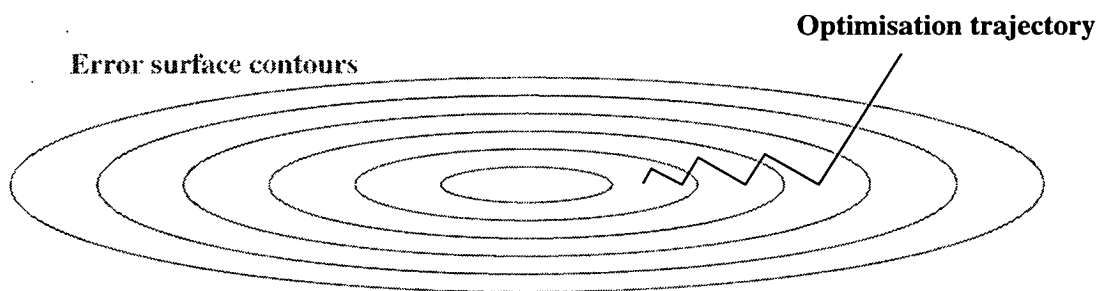
where  $\mathbf{w}$  is the parameter vector and  $\eta$  is the optimisation rate parameter.

This type of optimisation is extremely inefficient for error function minimisation due to the excessive number of function evaluations. It is also prone to oscillation along error surface valley's and consequently large numbers of iterations, even when the error surface is quadratic. A momentum term, which effectively acts like a smoother, can dampen these oscillations and improve convergence to the minimum. However, the value of the optimisation rate parameter and the amount of momentum is very important for fast convergence and is very much problem

dependent. Optimising a neural network with gradient descent with momentum is related to conjugate gradient minimisation (see Section A.3) with optimal values for optimisation rate and momentum set by the conjugate gradient algorithm.

There exist several enhanced gradient descent, such as *bold driver technique* or *quick-prop*, but these were not used as there exist many better optimisation techniques with better mathematical foundation.

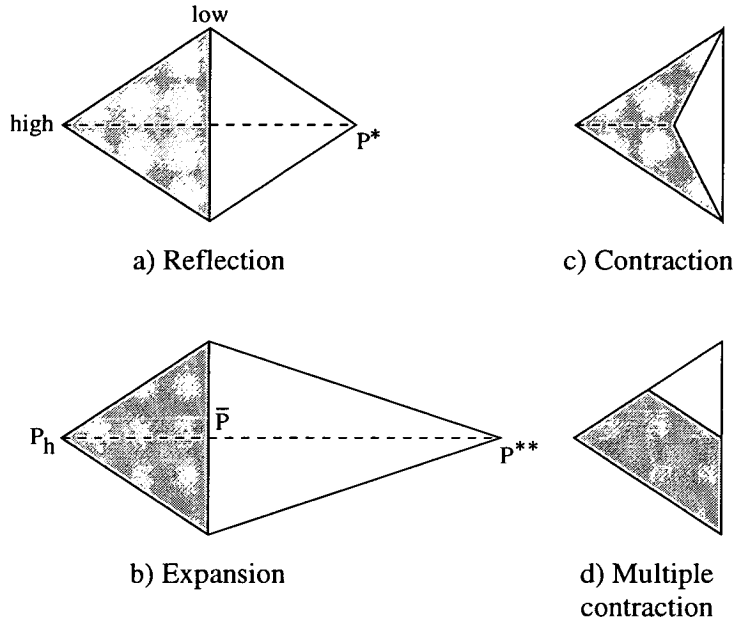
*Steepest descent* is worth mentioning though as it uses line searching much like conjugate gradient. Instead of using an optimisation rate parameter the new parameter vector is located at the minimum in the direction of the local negative error surface gradient. The new direction of search then proceeds in the direction of the gradient at the new parameter vector. However, this also suffers from the same oscillation problem described before as shown in Figure A-1.



**Figure A-1:** Steepest descent: Problems of oscillation.

## A.2 The Simplex Method

The downhill simplex method in multi-dimensions was proposed by Nelder and Mead [75,82] for function minimisation. This is a simpler algorithm than the conjugate gradient method in that no function derivatives need to be calculated, only pure function evaluations. However, it is not very efficient in the number of iterations required to reach a solution. But it is a way of providing a working solution without complex derivative calculations which may not even be available. Simplex is based on neither first or second derivatives. No assumptions are made about the surface except it is continuous and has a unique minimum in the area of the search. It performs well when the curvature of the error surface changes rapidly, when compared to other methods, but it may perform worse in the neighbourhood of the minimum. There are few multiplications and no divisions to be performed.



**Figure A-2:** Simplex: Possible steps in two dimensions.

A *simplex* is a geometrical figure consisting of  $N + 1$  vertices,  $P_0, P_1, \dots, P_N$ , and all their interconnecting line segments, where  $N$  is the dimension of the space. It is assumed that the simplex is of a finite volume, it is *non-degenerate*. Each point,  $P_i$  has an associated function value,  $y_i$ . The  $\bar{P}$  is the centroid of all the vertex positions not including the vertex with the highest function value. Equation A.1 represents a reflection as shown in Figure A-2a where  $\alpha$  is known as the *reflection coefficient*. Reflection occurs if the newly reflected point lies between the highest and lowest point in the simplex and also when an expansion fails. Equation A.2 represents an expansion as shown in Figure A-2a where  $\gamma$  is known as the *expansion coefficient*. An expansion will fail if, after a successful reflection, the simplex can not be extended any further in that direction without increasing the function value. Equation A.3 represents a contraction as shown in Figure A-2a where  $\beta$  is known as the *contraction coefficient*. A contraction will occur when the reflected point is higher than the current highest vertex. If the contraction fails the simplex is reduced in size towards the current minima as shown in Figure A-2d. The coefficients are set to be 1, 2, and 1/2 respectively [75].

$$P^* = (1 + \alpha)\bar{P} - \alpha P_h \quad (\text{A.1})$$

$$P^{**} = \gamma P^* + (1 - \gamma)\bar{P} \quad (\text{A.2})$$

$$P^{**} = \beta P_h + (1 - \beta)\bar{P} \quad (\text{A.3})$$

Simplex adapts to the local error surface, elongating down long inclined planes, changing directions, and contracting in the neighbourhood of a minimum. Initial size and orientation of the simplex will have an effect on the speed of convergence.

The stopping criteria compares the 'standard error' of the heights of the simplex vertices with a preset value. The success of the criteria depends on the simplex not becoming too small in relation to the curvature of the surface until final minima reached.

### A.3 Conjugate gradient optimisation

The concept of conjugate gradient optimisation has been around for about 30 years. Recently it has become one of the most popular, derivative-based, optimisation techniques for neural network optimisation, replacing the now outmoded gradient descent method [35,81,12].

The practical concept is similar to steepest descent with the exception that the new search direction is not necessarily orthogonal but conjugate to the previous search direction. In plain terms this means that the gradient vector along the new direction has a zero (to lowest order) component in a direction parallel to the previous search direction, as shown in Figure A-3. In this way the new direction does not interfere with previous minimisations.

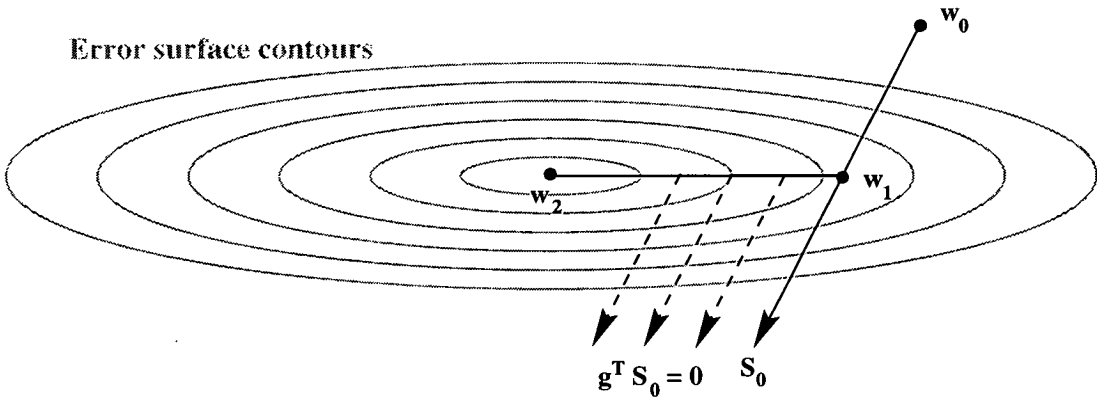


Figure A-3: Conjugate gradient optimisation.

When the error surface is quadratic and positive definite the Hessian can be used to determine the step sizes along the conjugate gradient directions. However, with a highly nonlinear error surface whereby local Hessians are not necessarily positive definite and also possibly compute intensive to generate, it is more usual to use a line minimisation to find the correct step size.

There are solutions to help include the Hessian such as scaled conjugate gradient but will not be considered in this thesis. The conjugate directions are generated through the *Polak-Ribiere* algorithm and the initial direction set equal to the local negative error surface gradient. The conjugate gradients are reinitialised every  $N$  steps, where  $N$  is the number of adaptive parameters.

In this thesis a golden search technique was used along with a simple bracketing algorithm as the line minimiser. However, it was found that many of the experiments that a considerable speed up could be achieved by replacing the golden search technique with Brent's algorithm which applies parabolic interpolation.

---

# Appendix B

## Publications

---

- M. Smart "Rotation invariant IR object recognition using adaptive kernel subspace projections with a neural network". In *Biological and Artificial Computation : From Neuroscience to Technology*, International Work-Conference on Artificial Neural Networks, Lanzarote, Volume 1240, 1028-1038. In print, 1997.
- R. Reavy, M. Smart and A. F. Murray "Identification of segmentation quality of real IR objects using feature novelty". To be published.
- M. Smart and A. F. Murray "A Multilayer Perceptron for Rotationally Invariant Feature Extraction and Classification". In *Proceedings of the SPIE International Conference on Applications and Science of Artificial Neural Networks II*, Florida, Volume 2760, 459-466. In print, 1996.
- M. Smart "Invariance with Neural Networks with an Application to Automatic Target Recognition". Presented at *NSYN Workshop: Practical Applications of Neural Networks*, Edinburgh. Presented, 1996.
- M. Smart and A. F. Murray "A Subspace Projection Approach for Rotationally Invariant ATR Feature Extraction". In *PhdEE: University of Edinburgh*, Volume 2. In print, 1995.
- M. Smart "A Multilayer Perceptron for Rotationally Invariant Feature Extraction and Classification". In *British Aerospace Systems and Equipment TN4291*. In print, 1995.
- M. Smart "Infrared image classification: Preliminary Investigation". In *British Aerospace Systems and Equipment TN4132*. In print, 1994.
- M. Smart "Noise in neural training and Infrared Image Classification: Database Preparation". In *British Aerospace Systems and Equipment TN3938*. In print, 1993.

Note: Only two publications are included in this appendix.



M. Smart. "Rotation invariant IR object recognition using adaptive kernel subspace projections with a neural network"

### **Rotation invariant IR object recognition using adaptive kernel subspace projections with a neural network**

Michael H. W. Smart

Dept. of Electrical Engineering,  
King's Buildings, University Of Edinburgh, Scotland.  
mhws@ee.ed.ac.uk

**Abstract.** This paper examines two techniques for rotation invariant, adaptive feature extraction and classification of infra red images using a feedforward neural network model. Both approaches use a set of adaptive kernels, or wavelets, to generate rotation invariant features for classification and allow for direct minimisation of a classification error criterion against the input images whilst maintaining a low dimensional parameter space. Each feature extraction parameter is estimated using errors backpropagated from the classification stage.

The first of the two methods uses complex kernels with adaptive radial polynomials. When combined with a magnitude nonlinearity in the first layer of the model they provide rotation invariant features for classification. However, there are several problems with this model which make it impractical. A second method provides a much simpler solution and uses the preprocessing technique of  $\theta$  normalisation with a standard adaptive feature extraction and classification model. Both of these methods have been tested on the difficult problem of discriminating between objects derived from a set of real infra red images. Results and discussion are provided in this paper.

## **1 INTRODUCTION**

There are many problems associated with the automatic recognition of objects derived from real infra red (IR) images. One of these problems is to maintain a constant misclassification rate regardless of either sensor or object rotation and many feature extraction based solutions have been proposed [10]. However, these methods often require extensive search techniques to determine a suitable subset of features for every new task. A more sensible approach is to optimise a combined feature extraction and classification network against an overall classification error criterion. Unfortunately due to the high dimensionality of the image space these networks tend to produce large numbers of adjustable parameters and with a finite data set this may lead to problems of underdeterminedness. They may also lack the desired invariance.

This paper examines two techniques for rotation invariant, adaptive feature extraction and classification using a feedforward neural network model whilst maintaining a relatively low dimensional parameter space. This is applied to a

specific problem of IR object recognition in which sensor rotation invariance and the ability of the system to easily adapt to new environments is essential.

One of the image databases used to test the model consists of 608 frames of seascape scenes, that were taken at a constant depression angle, from various coastal locations in South West England and contain 3 broad classes of large, man-made objects; namely sailboat, motor boat and buoy. These objects are detected and extracted using a Sobel based segmentation algorithm and spatially normalised to a size of 32x32 pixels, whilst maintaining object aspect ratio. This generated a database of 1609 objects, of which 738 were sailboats, 533 motor boats and 338 buoys.

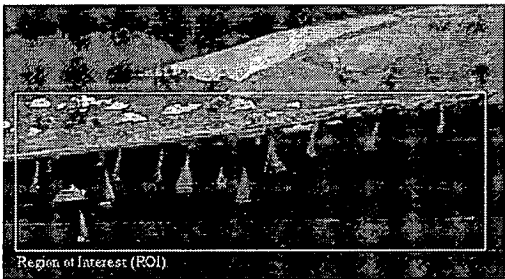


Fig. 1. Example from the seascape image database.

2 ADAPTIVE FEATURE EXTRACTION

The "super-wavelet" concept was introduced by *Szu et al.* as a combination of adaptive wavelet feature extraction and linear class discrimination [8] and has been applied successfully to problems of signal representation and classification. Many of the problems of feature selection were circumvented by this concept of a "super-wavelet" due to the direct adaptation of the feature extraction, whilst maintaining a controllable numbers of adjustable parameters.

The "super-wavelet" is a linear weighted sum of  $N$  adaptive wavelets, or kernels, which are shifted and dilated versions of a mother kernel,  $\psi$ . To classify a two-dimensional signal, such as an image  $f(x, y)$ , a linear discriminant of the form

$$z_k(f; \underline{\phi}) = w_{0k} + \sum_{j=1}^N w_{jk} \sum_x \sum_y f(x, y) \psi_j(x, y; \underline{\phi}_j) \tag{1}$$

can be implemented where  $z_k$  represents one of  $C$  classifier outputs and the full classification parameter vector,  $\Phi$ , is comprised of the weights and biases,  $w_{jk}$ , and the  $M$ -dimensional kernel parameter vectors,  $\phi_j$ . Hence, in the model there are  $MN + C(N + 1)$  adaptive parameters.

This combination of adaptive feature extraction followed by classification can be visualised, as with the macro-Gabor filter, as a two layer neural network with a linear hidden layer [?]. The initial layer of adaptive kernels provides a linear transformation of the image to a lower dimensional feature space and the output layer forms a linear discriminant. Hence, to avoid ill-conditioning during optimisation there can be no linear relationship between the kernel parameters' and  $\psi$ . This network can then be easily extended to a nonlinear classifier, such as a MultiLayer Perceptron (MLP), in the form

$$z'_k(f; \Phi) = w_{0k} + \sum_{j=1}^H w_{jk} \varphi \left( w_{0j} + \sum_{i=1}^N w_{ij} \sum_x \sum_y f(x, y) \psi_j(x, y; \phi_j) \right) \quad (2)$$

where  $H$  represents the number of hidden units with the  $\varphi$  nonlinearity.

Nonlinear optimisation of  $\Phi$  is performed using a conjugate gradient directed line searching technique to minimise an output classification error criterion,  $E$ . The error derivative,  $\partial E / \partial \phi_{jm}$ , for the linear network of Equation 1, can be easily derived over all the training patterns,  $t$ , to be

$$\sum_t \sum_k w_{jk} \frac{\partial E}{\partial z_k} \cdot \sum_x \sum_y f'(x, y) \frac{\partial \psi_j(x, y; \phi_j)}{\partial \phi_{jm}}.$$

As stated by Daugman [3] the resulting feature extractors,  $\psi_j$ , are required to be neither orthogonal ( $\langle \psi_j(x, y, x_0, y_0); \psi_k(x, y, x_0, y_0) \rangle \neq 0$  for all  $j \neq k$ ) nor complete in order to satisfy optimality according to  $E$  and the main consideration is the form of  $\psi$ .

Many authors use the Gabor wavelet as a suitable kernel and have successfully applied it to many problems including image representation [3], object detection [?] and character recognition [6]. The Gabor wavelet is given by

$$\psi(x, y; x_0, y_0, a, b, u_0, v_0) = \exp\{-\pi[(x - x_0)^2 a^2 + (y - y_0)^2 b^2]\} \cdot \exp\{-2\pi j[u_0(x - x_0) + v_0(y - y_0)]\} \quad (3)$$

and an example of the real part of a typical kernel is given in Figure 2. The Gabor kernel is a Gaussian, centred at  $(x_0, y_0)$  and with scaling values  $(a, b)$ , modulated with a complex exponential with spatial frequency  $(u_0^2 + v_0^2)^{1/2}$  and orientation  $\arctan(v_0/u_0)$ .

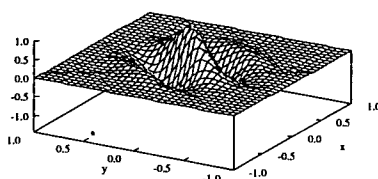


Fig. 2. Example of the real part of Gabor wavelet

### 3 ROTATION INVARIANCE (RI)

Rotation invariant classification is achieved if the *a posteriori* class dependent probability estimates of an object remain unaffected by image rotation. If the image is centred using central moments and scaled to be of unit radius it can be expressed in a polar coordinate system,  $f(\rho, \theta)$ , where  $\rho$  denotes radial direction and  $\theta$  angular direction. A rotation can then be expressed as a simple linear shift in the  $\theta$  direction by a constant  $\alpha$ , i.e.  $f(\rho, \theta + \alpha)$ . This paper concentrates purely on these simple in-plane rotations of an image.

Barnard and Casasent [1] identify three different neural based approaches to RI classification. **Invariance through training or regularisation:** The classification model is based on a training set that sufficiently covers the span of rotated images. Although simple, this method requires a significantly large database. **Invariance through structure:** The second approach is to encode RI properties within the model. A good example of this approach are high-order neural networks which can be made translation, rotation and scale invariant at order 3, by suitable choice of network parameters. However, these networks are sometimes impractical due to their size. **Invariance through preprocessing:** This is the most popular method and two particular approaches are considered in this paper; complex kernel feature extraction and  $\theta$  normalisation. Both methods allow for adaptive feature extraction.

#### 3.1 RI through complex kernel feature extraction

In their paper concerning Zernike circular polynomials Bhatia and Wolf [2] demonstrate that for a kernel to provide RI about the centre of mass of an object it must be of the form  $g(\rho)\exp(jm\theta)$  where  $m$  represents circular harmonic order and  $g(\rho)$  a radial polynomial.

Many authors use these complex kernels to generate RI features,  $d_i$ , as in

Equation 4 where  $*$  denotes the complex conjugate and  $|\cdot|$  complex magnitude.

$$d_i = \left| \int_0^1 \int_0^{2\pi} f(\rho, \theta) \psi_i^*(\rho, \theta) d\theta \rho d\rho \right| \quad (4)$$

The ability of this transform to achieve RI is demonstrated in Appendix A.1.

The choice of  $g(\rho)$  and  $m$  are obviously crucial for a minimising  $E$  and four types of kernel derived from Fourier-Mellin (FM), orthogonal Fourier-Mellin (OFM), Zernike (ZE) and pseudo-Zernike (PZ) moments have been found to work well [5, 9]. Fourier-Mellin moments use the kernel  $\psi_{is}(\rho, \theta) = \rho^s \exp(jm\theta)$  where in this paper  $s$  is integer valued. Sheng and Shen [5] derived OFM moments by orthogonalisation of the sequence  $1, \rho, \rho^2, \dots, \rho^n$ . This generates a set of orthogonal  $g(\rho)$  such that  $\psi_{in}(\rho, \theta) = \exp(jm\theta) \sum_{s=0}^n \beta_{ins} \rho^s$ . Two other sets of moments were discovered by a similar orthogonalisation of the sequences  $\rho^{|m|}, \rho^{|m|+2}, \dots, \rho^{|n|}$  and  $\rho^{|m|}, \rho^{|m|+1}, \dots, \rho^{|n|}$ . These are the ZE and PZ moments respectively [2]. In the same way as the OFM, the ZE and PZ kernels can be expressed as linear combinations of weighted natural powers of  $\rho$  but with  $\beta_{ins} = 0$  for  $s < m$ . More generally,

$$d_i = \left| \sum_{s=0}^n \beta_{ins} \int_0^1 \int_0^{2\pi} f(\rho, \theta) \rho^s e^{-jm\theta} d\theta \rho d\rho \right| = \left| \sum_{s=0}^n \beta_{ins} M_{sm} \right| \quad (5)$$

whereby suitable choice of  $\beta_{ins}$  can generate any of the required moments.

Teh and Chin [9] tested various image moments for information redundancy, noise sensitivity and image reconstruction capability. Of the moments examined Zernike had the best overall performance. However, the position of the ZE  $g(\rho)$  zeros, than say those of OFM, might not be so suitable for scale and RI classification [5]. Furthermore the number of feature moments used is often determined by a normalised reconstruction error and not directly by a classification error criterion. Smart *et al* [7] therefore attempts to combine feature extraction into an overall classification model by including  $\beta_{ins}$  as a classification parameter. This is an iterative method to automatically determine a suitable set of  $g(\rho)$ 's for a particular object recognition task. It can be constructed as in Equation 5 with complex magnitude nonlinearities in a single preprocessing layer before, for example, an MLP classifier.

However, there are several problems associated with this technique. Imagine the simple problem of optimising, with respect to a least squares error criterion, the network  $z = w |\beta M|$  where  $w$  is the output weight and  $\beta$  and  $M$  are as in Equation 5. The error surface for a problem using complex FM features derived from two distinct sets of rotated images is shown in Figure 3. The solution requires a positive output weight and the error surface in the negative region can cause line searching optimisation techniques to fail if the network is improperly initialised. Also, with a positive output weight the network can be expressed as  $z = |w\beta M|$  and hence there is ill-conditioning. A non-derivative based optimisation method, such as simplex, provides a working alternative and reasonable results can be achieved [7]. However, there are other factors that make the approach unattractive. These include the requirement of non-interference between

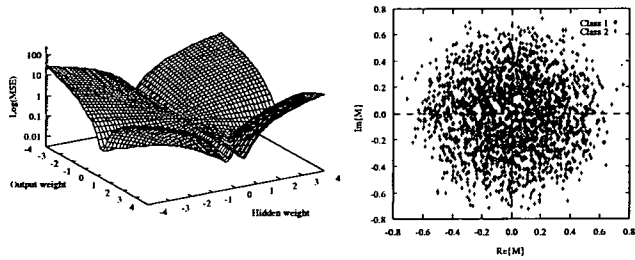


Fig. 3. Mean squared error (MSE) surface for a simple problem.

features using different  $m$  and a large number of parameters to a achieve satisfactory classification rate.

3.2 RI through  $\theta$  normalisation

A better approach is to transform the image such that RI is naturally incorporated into the new image. This allows for direct application of the standard adaptive feature extraction techniques discussed in Section 2 and can be achieved through  $\theta$  normalisation. This process normalises for rotations in an image by a linear shift in the  $\theta$  direction equal to the circular mean,  $\bar{\theta}$  [4] which is determined using  $\cos \bar{\theta} = C(\theta)/R(\theta)$  or  $\sin \bar{\theta} = S(\theta)/R(\theta)$  where  $R(\theta) = (C^2(\theta) + S^2(\theta))^{1/2}$  and

$$C(\theta) = \int_0^1 \int_0^{2\pi} \cos \theta f(\rho, \theta) \, d\theta \rho d\rho \text{ and } S(\theta) = \int_0^1 \int_0^{2\pi} \sin \theta f(\rho, \theta) \, d\theta \rho d\rho. \quad (6)$$

The new image,  $f(\rho, \theta + \bar{\theta})$ , is then invariant to the initial rotation of the image and this is proved in Appendix A.2. Also, by using a log-polar transform instead, scale invariance can also be achieved.

4 RESULTS

The  $\theta$  normalisation process is applied to the set of images, described in Section 1, which are mapped to a 20x72 polar coordinate system. The data is then randomly split into two sets, one for training consisting of 1000 patterns and one for testing of 500 patterns. This process is repeated 10 times for each experiment.

The first of these experiments is to test the standard RI feature extraction methods of moments discussed in Section 3. For each moment type a set of

features is classified using a least squares linear discriminant and the number within each set is determined by increasing moment order. The results are shown in Figure 4(a).

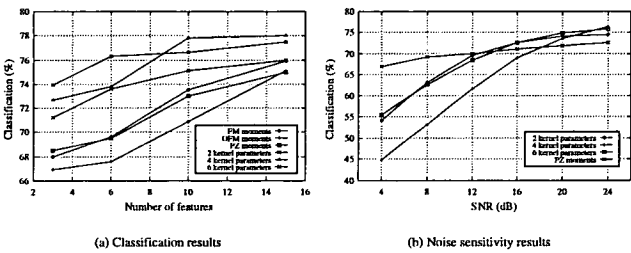


Fig. 4. Adaptive RI classification results.

This provides a benchmark for the adaptive feature extraction classifier given in Equation 1. The first adaptive experiment adapts the two positional kernel parameters ( $x_0, y_0$ ) only, with  $u = v = 0$ ,  $a = 2.50$  and  $b = 1.25$ . This is a simple Gaussian kernel. The following test optimises the four parameters of the Gaussian ( $a, b, x_0, y_0$ ), and in the final experiment using the real part of the Gabor wavelet as the kernel, all six kernel parameters are used. In each test the number of kernels,  $N$ , is varied and the optimisation process applied for 1000 iterations. Results are given in Figure 4(a) where each point represents the mean value, over 10 different random splits of the data, with a standard deviation of approximately 0.5%. The results indicate that for large values of  $N$  simple Gaussian kernels, with four adaptive parameters, will suffice.

One of the aspects of moment based features is their sensitivity to image noise and the pseudo-Zernike moments have been found to be less affected by noise than, for example, Fourier-Mellin or Zernike [9]. Figure 4(b) shows how the various adaptive models are affected by additive Gaussian noise in comparison with the PZ moments. It is clear that these models are more sensitive to noise than their moment based counterparts.

Figure 5 shows how the effective kernel linear classifier (i.e. the weighted sum of kernels) for the sailboat class changes during optimisation. The classifier has more kernels around the centre ( $\rho = 0$ ) and uses fewer, broader kernels towards the extremities of the image. These areas include the tops of sails and the bows of motor boats.

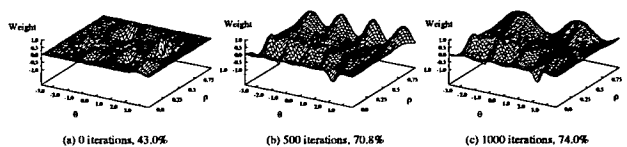


Fig. 5. Combined filter weights for sailboat class.

## 5 CONCLUSIONS

This paper has discussed two methods for rotation invariant adaptive feature extraction. The first method, using complex kernels, is difficult to optimise but a much simpler second approach of  $\theta$  normalisation allows for the direct application of standard adaptive techniques. These allow for optimisation of both feature extraction and classification while maintaining a low dimensional parameter vector.

The results on the seascape database show a significant improvement over the current fixed RI features, especially with a low number of features. However, they may appear disappointing with respect to an overall classification rate. This is because the data poses a difficult RI recognition problem with rotated boats often resembling motor boats or buoys. There also exists large within-class variations so there is a high probability of multimodal class distributions and the requirement of the nonlinear classifier as in Equation 2. Preliminary results are promising with 86.2% classification using the four parameter per kernel model with 6 nonlinear hidden units.

## 6 ACKNOWLEDGEMENTS

This work is being jointly funded by British Aerospace Systems and Equipment Ltd. (Applied Research project number 82140761) and EPSRC. I would also like to acknowledge Prof. Alan Murray for his support.

## References

1. E. Barnard and D. Casasent. "Invariance and neural nets". *IEEE Transactions on Neural Networks*, 2(5):498-508, 1991.
2. A. B. Bhatia and E. Wolf. "On the circle polynomials of Zernike and related orthogonal sets". *Proc. of the Cambridge Philosophical Society*, 50:40-48, 1954.
3. J. G. Daugman. "Complete Discrete 2-D Gabor Transforms by Neural Networks for Image Analysis and Compression". *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(7):1169-1179, 1988.



4. K. V. Mardia. "Statistics of Directional Data". Academic Press, 1972.
5. Y. Sheng and L. Shen. "Orthogonal Fourier-Mellin moments for invariant pattern recognition". *Journal of the Opt. Soc. of America (A)*, 11(6):1748-1757, 1994.
6. A. Shustorovich. "A Subspace Projection Approach to Feature Extraction: The Two-Dimensional Gabor Transform for Character Recognition". *Neural Networks*, 7(8):1295-1301, 1994.
7. M. H. W. Smart and A. Murray. "Multilayer Perceptron for Rotationally Invariant Feature Extraction and Classification". In *Proc. of the SPIE Int. Conf. on Applications and Science of ANN's II*, volume 2760, pages 459-466, 1996.
8. H. H. Szu, B. Telfer, and S. Kadambe. "Neural network adaptive wavelets for signal representation and classification". *Optical Engineering*, 31(9):1907-1916, 1992.
9. C. Teh and R. T. Chin. "On Image Analysis by the Method of Moments". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-10(4):496-513, 1988.
10. J. Wood. "Invariant Pattern Recognition: A Review". *Pattern Recognition*, 29(1):1-17, 1996.

## A PROOF OF RI

### A.1 RI through complex kernel feature extraction

RI is accomplished using features generated from the complex kernels,  $\psi(\rho, \theta) = g(\rho)\exp(jm\theta)$  and each RI feature,  $d_1$ , is determined by the equation

$$d_1 = \left| \int_0^1 \int_0^{2\pi} f(\rho, \theta) \psi^*(\rho, \theta) d\theta \rho d\rho \right| = \left| \int_0^1 \int_0^{2\pi} f(\rho, \theta) g(\rho) e^{-jm\theta} d\theta \rho d\rho \right|.$$

To prove the features are RI the effect of rotating the image by,  $-\alpha$ , is compared with the new feature,  $d_2$ , given by

$$d_2 = \left| \int_0^1 \int_0^{2\pi} f(\rho, \theta + \alpha) g(\rho) e^{-jm\theta} d\theta \rho d\rho \right|$$

and by letting  $\theta' = \theta + \alpha$  and knowing that  $|\exp(jm\alpha)| = 1$

$$d_2 = \left| e^{jm\alpha} \int_0^1 \int_0^{2\pi} f(\rho, \theta') g(\rho) e^{-jm\theta'} d\theta' \rho d\rho \right| = d_1.$$

### A.2 RI through $\theta$ normalisation

Assume the circular mean expressed in Equation 6 for an image  $f(\rho, \theta)$  is given by  $\bar{\theta}_1$  and for a rotated version of the same image,  $f(\rho, \theta + \alpha)$ , by  $\bar{\theta}_2$ . Then for RI  $\bar{\theta}_2$  must equal  $\bar{\theta}_1 - \alpha$ . For the rotated image

$$C^2(\theta) = \left[ \int_0^1 \int_0^{2\pi} \cos\theta f(\rho, \theta + \alpha) d\theta \rho d\rho \right]^2$$

and by letting  $\theta' = \theta + \alpha$  it can be seen that

$$C^2(\theta') = \left[ \int_0^1 \int_\alpha^{2\pi+\alpha} \cos(\theta' - \alpha) f(\rho, \theta') d\theta' \rho d\rho \right]^2$$

$$\begin{aligned}
&= \left[ \cos \alpha \int_0^1 \int_\alpha^{2\pi+\alpha} \cos \theta' f(\rho, \theta') d\theta' \rho d\rho + \sin \alpha \int_0^1 \int_\alpha^{2\pi+\alpha} \sin \theta' f(\rho, \theta') d\theta' \rho d\rho \right]^2 \\
&= \cos^2 \alpha \left[ \int_0^1 \int_\alpha^{2\pi+\alpha} \cos \theta' f(\rho, \theta') d\theta' \rho d\rho \right]^2 + \sin^2 \alpha \left[ \int_0^1 \int_\alpha^{2\pi+\alpha} \sin \theta' f(\rho, \theta') d\theta' \rho d\rho \right]^2 \\
&+ 2 \cos \alpha \sin \alpha \left[ \int_0^1 \int_\alpha^{2\pi+\alpha} \cos \theta' f(\rho, \theta') d\theta' \rho d\rho \right] \left[ \int_0^1 \int_\alpha^{2\pi+\alpha} \sin \theta' f(\rho, \theta') d\theta' \rho d\rho \right]
\end{aligned}$$

Similarly,

$$\begin{aligned}
S^2(\theta') &= \cos^2 \alpha \left[ \int_0^1 \int_\alpha^{2\pi+\alpha} \sin \theta' f(\rho, \theta') d\theta' \rho d\rho \right]^2 + \sin^2 \alpha \left[ \int_0^1 \int_\alpha^{2\pi+\alpha} \cos \theta' f(\rho, \theta') d\theta' \rho d\rho \right]^2 \\
&- 2 \cos \alpha \sin \alpha \left[ \int_0^1 \int_\alpha^{2\pi+\alpha} \cos \theta' f(\rho, \theta') d\theta' \rho d\rho \right] \left[ \int_0^1 \int_\alpha^{2\pi+\alpha} \sin \theta' f(\rho, \theta') d\theta' \rho d\rho \right]
\end{aligned}$$

Now  $R^2(\theta') = C^2(\theta') + R^2(\theta')$  such that

$$\begin{aligned}
R^2(\theta') &= \left[ \int_0^1 \int_\alpha^{2\pi+\alpha} \cos \theta' f(\rho, \theta') d\theta' \rho d\rho \right]^2 + \left[ \int_0^1 \int_\alpha^{2\pi+\alpha} \sin \theta' f(\rho, \theta') d\theta' \rho d\rho \right]^2 \\
&= \left[ \int_0^1 \int_0^{2\pi} \cos \theta' f(\rho, \theta') d\theta' \rho d\rho \right]^2 + \left[ \int_0^1 \int_0^{2\pi} \sin \theta' f(\rho, \theta') d\theta' \rho d\rho \right]^2 = R^2(\theta) \quad (7)
\end{aligned}$$

So  $R(\theta)$  is invariant to changes in image rotation but what about  $C(\theta)$ ?

$$\begin{aligned}
C(\theta') &= \int_0^1 \int_\alpha^{2\pi+\alpha} \cos(\theta' - \alpha) f(\rho, \theta') d\theta' \rho d\rho \\
&= \cos \alpha \int_0^1 \int_0^{2\pi} \cos \theta' f(\rho, \theta') d\theta' \rho d\rho + \sin \alpha \int_0^1 \int_0^{2\pi} \sin \theta' f(\rho, \theta') d\theta' \rho d\rho
\end{aligned}$$

Combining this with  $R(\theta')$  from equation 7 and letting  $\theta' = \theta$  it can be seen that

$$\cos \bar{\theta}_2 = \frac{C(\theta')}{R(\theta')} = \cos \alpha \cos \bar{\theta}_1 + \sin \alpha \sin \bar{\theta}_1 = \cos(\bar{\theta}_1 - \alpha)$$

and hence

$$\bar{\theta}_2 = \bar{\theta}_1 - \alpha$$

M. Smart and A. F. Murray. "Multilayer perceptron for rotationally invariant feature extraction and classification"

## Multilayer perceptron for rotationally invariant feature extraction and classification

Michael H.W. Smart

Edinburgh University, Department of Electrical Engineering  
Mayfield Road, Edinburgh EH9 3JL

Alan F. Murray

Edinburgh University, Department of Electrical Engineering  
Mayfield Road, Edinburgh EH9 3JL

### ABSTRACT

In this paper we introduce a technique for incorporating adaptive, rotationally invariant (RI), feature extraction into the initial layer parameters of a multilayer perceptron (MLP) for classifying real infra-red (IR) imagery. Feature extraction parameters are not usually estimated directly due to their high dimensionality but it is possible to reduce the dimensionality by constraining these parameters to a feature subspace where the parameters are restricted to a continuous RI generating functional form (e.g. a circularly symmetric radial polynomial transform.) The lower dimensional function parameters and the classification parameters can then be estimated simultaneously to minimise an overall classification error criterion. This can be considered as an extension of previous work by other authors where non-RI filter parameters, such as Gabor filter directional selectivity, were successfully tuned for feature extraction.

**Keywords:** rotational invariance, multilayer perceptron, misclassification rate, infra-red imagery

### 1. INTRODUCTION

An important aspect of automatic target recognition (ATR) is the location and identification of possible targets in a scene, irrespective of sensor position or rotation. In this paper we investigate the possibility of adaptively tuning rotation invariant (RI) feature generating kernels in order to minimise an overall classification error criterion. These RI generating kernels are based on circular Fourier and radial Mellin transforms described in a polar coordinate system:

$$\int_0^1 \int_0^{2\pi} f(\rho, \theta) \rho^s \exp(-jm\theta) d\theta \rho d\rho, s \geq 0, m = 0, \pm 1, \pm 2, \dots, \pm \infty \quad (1)$$

where  $f(\rho, \theta)$  is an object image defined over the unit circle ( $0 \leq \rho \leq 1$ ),  $m$  is the circular harmonic order and  $s$  the transform order.<sup>5,14</sup>

Figure 1 shows a typical ATR system on which we shall concentrate on the *feature extraction* and *classification* functional units. Feature extraction is a form of linear or non-linear mapping that attempts to retain discriminational information whilst projecting data into a lower dimensional feature space. This both reduces computational complexity and generally allows more accurate parameter estimates with a limited set of observations. The feature extraction mapping can not usually be directly estimated with respect to an overall classification error criterion. This is often due to the high dimensionality of the object images producing the possibility that the number of independent parameters in the model significantly exceeding the limited number of training observations. Although several authors have suggested using techniques such as weight decay, cascade correlation or shared weights as a solution to this problem<sup>6,8,9</sup> a more general solution is to utilise a fixed set of feature extractors such as Karhunen-Loève, Hadamard, Harr, Fourier, Gabor, and singular value decomposition (SVD).<sup>18</sup>

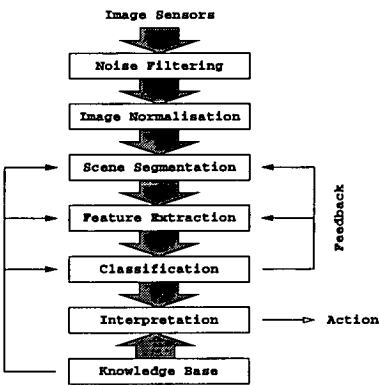


Figure 1: Typical ATR system.

1.1. Rotational invariance

Rotational invariance requires that features, and consequently object classifications, generated by a spatial mapping remain unaffected under pure rotations of the input image.<sup>2</sup> To achieve rotationally invariant pattern recognition Barnard and Casasent<sup>3</sup> identified three basic approaches, with respect to neural network based systems:

- Development of a classification model based on a training set that contains a sufficient number of examples of rotated images. Although simple, this method requires a significantly large database.
- The next approach is to hard-wire RI properties within the model. A good example of this approach are high-order neural networks (HONN),

$$x_i = \sum_{i_1} \sum_{i_2} \dots \sum_{i_p} \lambda_{i_1 i_2 \dots i_p} f_{i_1} \dots f_{i_p}, \tag{2}$$

which can be made translation, rotation and scale invariant at order 3, by suitable choice of the parameters,  $\lambda$ .<sup>11</sup>

- The final method, and possibly most popular, is the generation of RI features through pre-processing. We shall consider one form of pre-processing whereby each feature,  $x_i$ , is generated through a fixed complex function, or *kernel*,  $W_i(\rho, \theta)$ . The required invariance properties, such as insensitivity to sensor rotation or tilt, are incorporated into this initial pre-processing.

## 2. FOURIER-MELLIN AND ZERNIKE MOMENTS

In their paper concerning Zernike circular polynomials Bhatia and Wolf<sup>4</sup> demonstrated that there exist an infinite number of complete sets of polynomials which are orthogonal for the interior of the unit circle. They also showed that for a polynomial, or *kernel*,  $W(\rho \cos \theta, \rho \sin \theta)$  to be invariant in form about the origin it must be of the form  $g(\rho) \exp(jm\theta)$  where  $m$  is the circular harmonic order and  $g(\rho)$  a radial polynomial.

Many authors have proceeded to utilise these kernels in order to generate sets of rotationally invariant features,  $x_i$ , from centered and scaled polar images  $f(\rho, \theta)$ , as shown in equation 3 where  $*$  denotes the complex conjugate and  $|\cdot|$  complex magnitude.

$$x_i = \left| \int_0^1 \int_0^{2\pi} f(\rho, \theta) W_i^*(\rho, \theta) d\theta \rho d\theta \right| \quad (3)$$

The rotationally invariant property of this transform can easily be demonstrated by replacing  $f(\rho, \theta)$  by  $f(\rho, \theta + \phi)$  where  $\phi$  represents a sensor rotation away from the horizontal, and factoring out the term  $|\exp(jm\phi)| = 1$ .

The choice of the radial polynomial and circular harmonic order are obviously fundamental to the misclassification rate. We shall concentrate on four types of kernel derived from Fourier-Mellin (FM), orthogonal Fourier-Mellin (OFM), Zernike (ZE) and pseudo-Zernike (PZ) moments.<sup>15,17</sup> The kernel used to generate Fourier-Mellin moments is given by  $W_{is}(\rho, \theta) = \rho^s \exp(jm\theta)$ .  $s$  is usually complex valued ( $s = j\omega$ ) but we shall consider only integer values for  $s$ . Fourier-Mellin moments with integer valued  $s$  are often called rotational moments. Sheng and Shen<sup>15</sup> derived a new set of moments for invariant pattern recognition called orthogonal Fourier-Mellin moments by the Gram-Schmidt orthogonalisation of the sequence  $1, \rho, \rho^2, \dots, \rho^n$ . This generates a set of orthogonal radial polynomials such that  $W_{in}(\rho, \theta) = \exp(jm\theta) \sum_{s=0}^n \alpha_{ins} \rho^s$ . Two other sets of moments, derived from the work of Frits Zernike on optical aberrations and diffraction, were discovered by the orthogonalisation of the sequences  $\rho^{|m|}, \rho^{|m|+2}, \dots, \rho^{|n|}$  and  $\rho^{|m|}, \rho^{|m|+1}, \dots, \rho^{|n|}$ . These are called Zernike and pseudo-Zernike moments respectively.<sup>4</sup> Thus in the same way as the OFM the Zernike kernels can be expressed as a linear combination of weighted natural powers of  $\rho$  but with  $\alpha_{ins} = 0$  for  $s < m$ . More generally we can write

$$x_i = \left| \sum_{s=0}^n \alpha_{ins} \int_0^1 \int_0^{2\pi} f(\rho, \theta) \rho^s \exp(-jm\theta) d\theta \rho d\theta \right| \quad (4)$$

whereby suitable choice of  $\alpha_{ins}$  we can generate any of the required moments. It is worth noting that both the Zernike, pseudo-Zernike and orthogonal Fourier-Mellin are derived from the more general Jacobi polynomials. Examples of the radial polynomials are provided in Figure 2.

In a study of image moments Teh and Chin<sup>17</sup> tested various types of moments including Zernike, pseudo-Zernike and Fourier-Mellin for information redundancy, noise sensitivity and image reconstruction capability. Of all these moments Zernike moments had the best overall performance. However, it has been suggested that due to the positioning of ZM radial polynomial zeros more towards the unit circle than say those of OFM polynomials ZM might not be so suitable for scale and rotation invariant classification.<sup>15</sup> Furthermore the choice of the number of moments used to perform the necessary feature extraction is often guided by a normalised reconstruction error and not by an overall classification error criterion. In this paper we examine the possibility of including the feature extraction into an overall classification model by kernel adaptation by including  $\alpha_{ins}$  as a classification parameter.

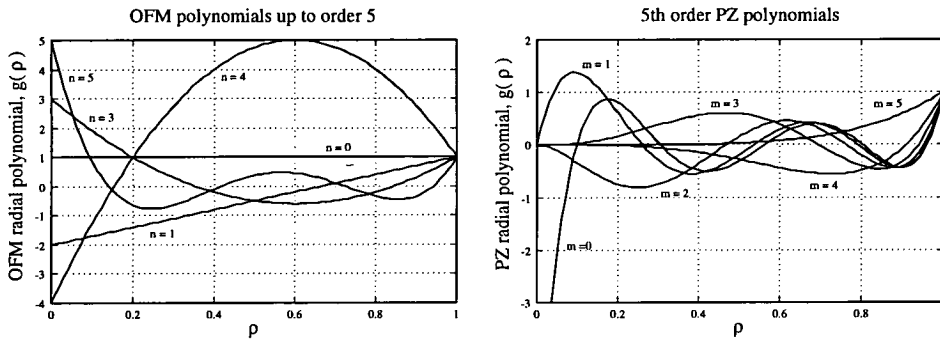


Figure 2: Examples of radial polynomials

### 3. CLASSIFICATION AND KERNEL ADAPTATION

Allocation of an arbitrary unclassified feature vector,  $\underline{x}$ , to a member of a predefined set,  $\omega_k$  ( $k = 1..N_c$ ), or class, is often achieved by comparison of the class *a posteriori* probabilities,  $P(\omega_k|\underline{x})$ .<sup>10</sup> However, the form of the class conditional probability density functions are often unknown. Thus we are left a task of nonparametrically estimating  $c$  discriminant functions,  $y_k(\underline{x})$ , such that  $y_i(\underline{x}) > y_j(\underline{x})$  for  $i \neq j$  given  $\underline{x}$  is of class  $\omega_i$ . Although nominally a parametric model the multilayer perceptron<sup>12</sup> (MLP), given in equation 5, provides a flexible method for parameterising a fairly general non-linear set of these discriminant functions. In fact MLP's are *universal approximators* in that given sufficient complexity and data they can approximate virtually any function.<sup>19</sup>

$$y_k(\underline{x}; \underline{\beta}) = \alpha_k + \sum_j^H w_{jk} \psi(\alpha_j + \underline{w}_j \cdot \underline{x}) \quad (5)$$

The approximating function is controlled by a vector of parameters,  $\underline{\beta}$ , comprising of a set of weights,  $\underline{w}$ , and biases,  $\underline{\alpha}$ , and the hidden layer activation function,  $\psi$ , is usually the logistic function

$$\psi(z) = 1/(1 + e^{-z}). \quad (6)$$

In this paper we form a least squares estimate,  $\hat{\underline{\beta}}$ , of the true model parameter vector,  $\underline{\beta}$ , using a conjugate gradient, iterative, local optimisation technique.<sup>1</sup> Model complexity can be controlled through varying the number of hidden nodes,  $H$ , as well as through various standard regularisation techniques.

The feature vector,  $\underline{x}$ , will be generated via equation 4 and can be considered as an initial pre-processing layer to the MLP model. The kernel parameters,  $\alpha_{ins}$ , as described in the previous section, can be fixed such as to implement a specific image moment but we were interested to see whether kernel parameters could be included into the extended MLP model such that the classification error minimisation was performed over a new parameter vector,  $\underline{\beta}'$ .

4. EXPERIMENTS

In order to test the adaptive kernel algorithm a series of simple artificial image object databases were created. The images were of the form  $f(\rho, \theta) = a_1(\rho)\cos(2\theta + \phi) + a_2(\rho) + \eta(\rho, \theta)$  where  $\eta(\rho, \theta)$  represents a additive, zero mean, white noise process. Various fixed kernel feature extraction methods were then tested against the adapted kernel method ( $m = 2$ ). Some results, using both an MLP and a K-nearest neighbour<sup>10</sup> (K=7) classifier, from one such test are recorded in Table 1.

Kernel	MLP (%) (Standard error)	KNN (%) (Standard error)
PZ	20.9 (1.48)	20.2 (1.46)
OFM	22.2 (1.52)	23.2 (1.55)
FM	23.6 (1.75)	23.9 (1.81)
Adapted	19.6 (1.32)	N/A

Table 1: Classification error rates for an artificial problem

Although the results are promising and demonstrate that kernel adaptability is feasible there does appear to be a problem with either local minima or the learning mechanism which occasionally causes the formation of inappropriate kernels, although the reasoning behind this phenomenon has yet to be confirmed.

The adaptive kernel algorithm was then tested on real, 8 bit greyscale, (8-12 $\mu$ m) infra-red (IR) seascape imagery. A typical 512x512 pixel IR seascape scene, with zero sensor rotation, is shown in Figure 3 with classification results from a Zernike feature extractor combined with a MLP classifier.

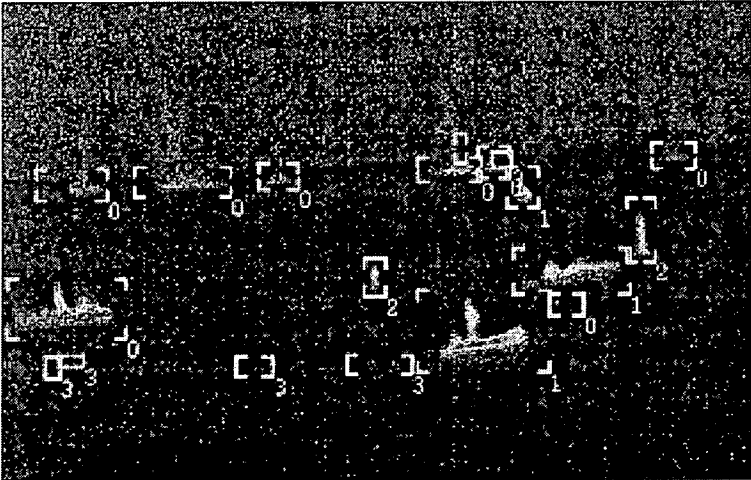


Figure 3: IR seascape scene with  $\phi = 0$

The image sequences contain seascape objects, including sailboat, motor boat, buoy, and clutter. Objects are identified and segmented utilising a standard Sobel edge detector, threshold and edgwalker in order to generate a database of 4000 objects. Once collated these were made translation and scale invariant by using low order image moments.<sup>16</sup> Three binary examples of the seascape classes are shown in Figure 4. The database was then divided into 3 separate training, validation and testing sets.

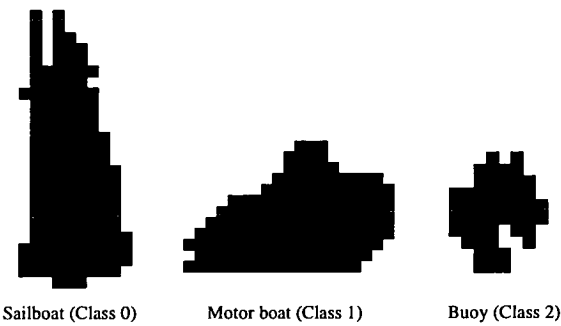


Figure 4: Typical binary objects with  $\phi = 0$

These objects contain a considerable amount of directional discriminatorial information provided that the sensor remains horizontal,  $\phi = 0$ . In this circumstance a non-RI feature extractor such as a Gabor filter<sup>7</sup> will classify better than, for example, a RI Zernike based system. However, as clearly demonstrated in Figure 5, only a relatively small tilt in the images that generate the test set object database (system trained with  $\phi = 0$ ) is required to incur a notable increase in the misclassification rate. As sensor rotation is expected in the project the use of an RI feature extractor is justified.

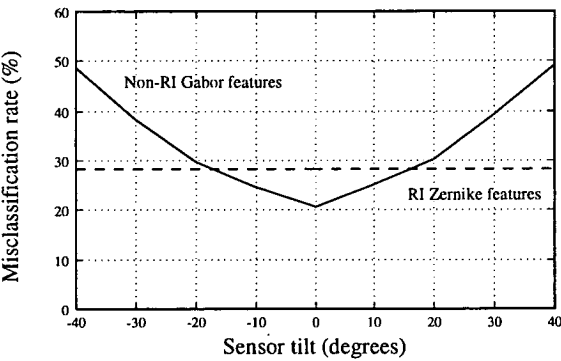


Figure 5: Effect of sensor tilt

Initially fixed RI kernels were used to generate features for the MLP classifier and estimates of  $\beta$  were formed for each type of kernel. Experiments were repeated to measure the statistical significance of any increases in



classification performance. Some results are provided in Table 2 including complex moments<sup>17</sup> (CM) and a RI correlation filter<sup>13</sup> (CHF).

Kernel	Features	KNN (%) (Standard error)	MLP (%) (Standard error)
FM	20	25.5 (1.2)	24.4 (1.5)
OFM	20	22.6 (1.3)	22.1 (1.3)
PZ	18	21.2 (1.6)	19.8 (1.2)
ZM	18	22.9 (1.3)	21.9 (1.4)
CM	15	26.0 (1.5)	25.3 (1.1)
CHF	20	21.3 (1.6)	20.1 (1.3)

Table 2: Classification results using fixed RI kernels.

A set of parameter estimates of  $\beta'$  were then formed to investigate whether any improvement in classification performance could be achieved with the new model. The adaptive kernel algorithm produced a misclassification rate of 19.2% (1.1). The kernel adaptation method has improved over certain types of moments but has provided only comparable performance with the pseudo-Zernike and correlation based classifiers.

## 5. CONCLUSIONS

In this paper we have attempted to demonstrate the possibility of RI kernel adaptability based around Fourier-Mellin moments using a MLP. We have shown that it can be successfully applied to a real IR problem although in the seascape database there was no significant decrease in the misclassification rate when compared to the best fixed RI kernel. However, it does provide a method of automatically generating RI kernels that are related to an overall classification error criterion. Further work will investigate the problem of local minima and the use of regularisation terms in the optimisation algorithm to reduce kernel correlation.

## 6. ACKNOWLEDGEMENTS

This work is being jointly funded by British Aerospace Systems and Equipment Ltd., Plymouth, England (Applied Research project number 82140761) and the Engineering and Physical Sciences Research Council.

## 7. REFERENCES

- [1] P. R. Aaby and M. A. H. Dempster. "Introduction to Optimization Methods". Chapman and Hall, 1978.
- [2] H. H. Arsenault, Y.-H Hsu, and K. Chalasinska-Macukow. "Rotation-invariant pattern recognition". *Optical Engineering*, 23:705, 1984.
- [3] E. Barnard and D. Casasent. "Invariance and neural nets". *IEEE Transactions on Neural Networks*, 2(5):498-508, 1991.
- [4] A. B. Bhatia and E. Wolf. "On the circle polynomials of Zernike and related orthogonal sets". *Proceedings of the Cambridge Philosophical Society*, 50:40-48, 1954.

- [5] D. Casasent and D. Psaltis. "Position, rotation and scale invariant optical correlation". *Applied Optics*, 15:1795-1799, 1976.
- [6] Y. Le Cun, J. S. Denker, and S. Solla. "Optimal brain damage". In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 598-605, San Mateo, CA, 1990. Morgan Kaufmann publishers.
- [7] J. G. Daugman. "Complete Discrete 2-D Gabor Transforms by Neural Networks for Image Analysis and Compression". *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(7):1169-1179, 1988.
- [8] S. E. Fahlman and C. Lebiere. "The cascade-correlation learning architecture". In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 524-532, San Mateo, CA, 1990. Morgan Kaufmann publishers.
- [9] A. Krogh and J. Hertz. "A simple weight decay can improve generalisation". In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 950-957, San Mateo, CA, 1992. Morgan Kaufmann publishers.
- [10] G. J. McLachlan. *"Discriminant Analysis and Statistical Pattern Recognition"*. Wiley and Sons, 1992.
- [11] S. J. Perantonis and P. J. G. Lisboa. "Translation, Rotation, and Scale Invariant Pattern Recognition by Higher-Order Neural Networks and Moment Classifiers". *IEEE Transactions on Neural Networks*, 3(2):241-251, March 1992.
- [12] B. D. Ripley. *"Pattern Recognition and Neural Networks"*. Cambridge University Press, 1996.
- [13] G. F. Schils and D. W. Sweeney. "Rotationally invariant correlation filtering". *Journal of the Optical Society of America (A)*, 2(9):1411-1418, September 1985.
- [14] Y. Sheng. "Fourier-Mellin spatial filters for invariant pattern recognition". *Optical Engineering*, 28(5):494-500, 1989.
- [15] Y. Sheng and L. Shen. "Orthogonal Fourier-Mellin moments for invariant pattern recognition". *Journal of the Optical Society of America (A)*, 11(6):1748-1757, 1994.
- [16] M. R. Teague. "Image analysis via the general theory of moments". *Journal of the Optical Society of America*, 70(8):920-930, 1980.
- [17] C. Teh and R. T. Chin. "On Image Analysis by the Method of Moments". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-10(4):496-513, July 1988.
- [18] C. W. Therrien. *"Decision Estimation and Classification"*. Wiley and Sons, 1989.
- [19] H. White. *"Artificial Neural Networks: Approximation and Learning Theory"*. Blackwell, 1992.